N A S A   C O N T R A C T O R
R E P O R T

NASA CR-132732-2

NASA CR-132732-2

(NASA-CR-132732-2) ADDITION OF FLEXIBLE BODY OPTION TO THE TOLA COMPUTER PROGRAM. PART 2: USER AND PROGRAMMER DOCUMENTATION (NASA) 205 p HC A10/MF A01 CSCL 01C

N85-27856

Unclas
G3/05 21527

ADDITION OF FLEXIBLE BODY OPTION TO THE TOLA COMPUTER PROGRAM

Part II - User and Programmer Documentation

By J. W. Dick and B. J. Benda

FOR EA[illegible]

JUL 1985
PUBLICLY RELEASED ON

Pre[illegible]
MCD[illegible]
St. Louis, Missouri 63166 (314) 232-0232

Date for general release: [redacted] APRIL - 1985

for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • OCTOBER 1975

NASA CR-132732 -2

ADDITION OF FLEXIBLE BODY OPTION TO THE TOLA COMPUTER PROGRAM

Part II - User and Programmer Documentation

By J. W. Dick and B. J. Benda

Prepared Under Contract NAS 1-13259

by

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY — EAST

St. Louis, Missouri

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

ABSTRACT

User and programmer oriented documentation for the flexible body option of the Takeoff and Landing Analysis (TOLA) computer program are provided in this report. The user information provides sufficient knowledge of the development and use of the option to enable the engineering user to successfully operate the modified program and understand the results. The programmer's information describes the option structure and logic enabling a programmer to make major revisions to this part of the TOLA computer program.

TABLE OF CONTENTS

LIST OF FIGURES

Pages

i through viii
1 through 60
A1 through A98
B1 through B2
C1 through C4
D1 through D14
E1 through E20

1. INTRODUCTION

The purpose of this report is to summarize the flexible body option of the airplane Takeoff and Landing Analysis (TOLA) computer program and explain how to use this option in conjunction with the overall program. The flexible body option was developed by McDonnell Douglas Astronautics Company-East under NASA Contract NAS1-13259 for investigation of aircraft response in a ground operating environment.

The original TOLA computer program developed by Air Force Flight Dynamics Laboratory personnel provides a complete simulation of the takeoff and landing of a rigid airplane. The flexible body option provides the additional capability of predicting the total motion of specified points on the aircraft including elastic airframe effects.

It should be emphasized that the flexible body option is but a part of the overall TOLA computer program. Successful operation of the program depends on proper input of all required data and interpretation of the resulting output. Much of these data are associated with the existing rigid body version of TOLA. It is not the purpose of this document to define these data, for this is done in Reference 1. The user should not attempt to execute the TOLA computer program without first becoming familiar with the information contained in this reference.

Programming information on the structure and flow logic of the flexible body option is also contained in this report. Additional information on the structure of the entire TOLA computer program is given in Reference 2.

2. PROGRAM DEVELOPMENT

2.1 Program Description and Capabilities - The flexible body option version of TOLA provides the capability to predict the total motion of an aircraft in the ground operating environment including airframe elastic effects and landing gear dynamics. It can simulate any conventional aircraft having up to five landing gears and four engines with the airframe considered either rigid or flexible. In the flexible airframe option, the flexibility is represented by superposition of the free-free vibratory modes on the rigid body motion. From one to twenty modes may be included to represent this airframe flexibility. The dynamic effects of a maximum of five independent landing gears can also be simulated. The landing gear modelled in the program is a double air chamber oleo strut with balloon tires, similar to that used on the C-5 aircraft (Figure 2-1). Each of the struts must lie in a plane parallel to the aircraft plane of symmetry, but the strut axis may be non-perpendicular to the longitudinal aircraft axis.

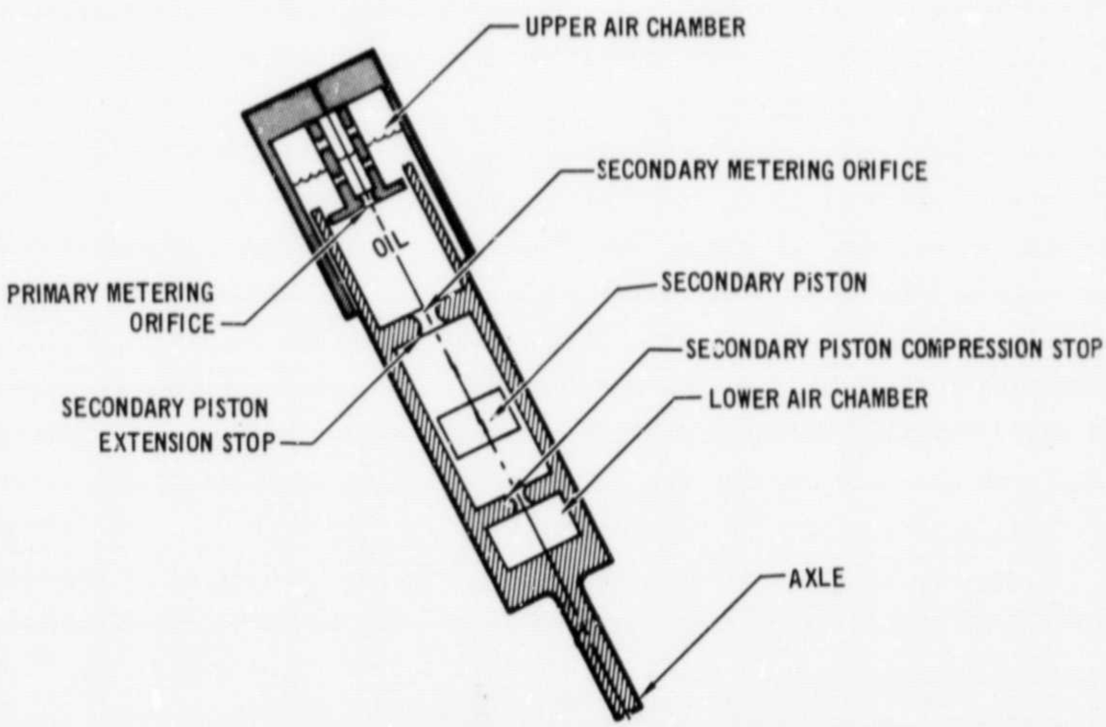


FIGURE 2-1 LANDING GEAR STRUT CONFIGURATION

PRECEDING PAGE BLANK NOT FILMED

The simulated aircraft, as represented by the airframe rigid and flexible body equations of motion and the strut, strut secondary piston and wheel equations of motion, may be subjected to time varying forces. The forces considered in the problem include engine thrust, aerodynamic loads including ground effects, drag chute forces, landing gear tire-ground reaction and braking forces. These forces may be varied and/or staged within the aircraft's capability by the maneuver logic and autopilot simulation.

The function of the autopilot simulation is to specify the timing and magnitude of the aircraft's control system inputs required for the desired aircraft performance during glide slope, flare, landing roll and takeoff roll. The maneuver logic utilizes information on the state of the aircraft obtained from solution of the airplane equations of motion to compute state errors. If necessary, it defines maneuvers required to maintain proper aircraft attitude by specifying parameters such as angle of attack, roll angle, thrust level and braking values. The maneuver logic also controls the staging of such events as spoiler activation, kill power, thrust reversal, drag chute deployment and brake activation in both normal and unusual takeoff or landing situations. The unusual situations include engine failure, brake failure or tire blow-out.

2.2 Development of Equations of Motion - To determine the response of an aircraft under any of the conditions described above, the equations of motion for that aircraft must be solved. As complex as the various options in the TOLA computer program may seem, they simply calculate their particular influence on these equations. The flexible body option is no exception. To understand this option and its effect on the overall program, its contributions to the existing equations of motion must first be understood. Since the equations which define the effects of a flexible airframe cannot be uncoupled from the overall problem, this section will give a general development of the total aircraft equations of motion including flexibility effects.

In developing the airplane equations of motion, the airplane main body is defined as the airplane less its landing gears. As such, the airplane is represented by K+1 bodies. The aircraft main body is considered the 0^{th} body while a typical landing gear is considered the k^{th} body with K being the total number of gears. Although the landing gears are considered rigid, they are allowed to move relative to the main body.

The coordinate systems used in the development of the equations of motion are shown in Figure 2-2. The inertial coordinate system (X_g, Y_g, Z_g) is fixed relative to the earth's local acceleration gravity vector. The body coordinate system (X_o, Y_o, Z_o) moves with the airplane and is fixed at the center of mass. The strut coordinate system (X_k, Y_k, Z_k) moves with the airplane and is located relative to the body coordinate system by the vector $(\overline{R}_k)_o$ and the angle Θ_k. This axes system is aligned such that the direction of gear movement is along the Z_k vector.

The dynamic motion of the main body is described using the normal mode method. In this method the main body motion is approximated by the combination of a limited number of vibratory modes plus the six rigid body modes. The main body's flexibility is represented by its free-free (unrestrained) vibratory modes. The rigid body modes are assumed to be the three translational displacements defining the airplane center of mass and three angular displacements defined in the body coordinate system.

In developing the airplane equations of motion, expressions defining the motion of an arbitrary point located on the landing gear and/or main body were obtained. These were used to evaluate the kinetic and potential energy of the airplane. The equations of motion were obtained by applying the Lagrangian equations to these energy expressions.

Using Figure 2-3, the total displacement of a point i is defined as

$$\overline{\rho}_{ki} = \overline{R} + (\overline{R}_k)_o + \overline{r}_{ki} \tag{2-1}$$

$\overline{\rho}_{ki}$ = position vector of point i relative to the inertial coordinate system.

$\overline{R}$ = position vector of reference point on main or 0^{th} body relative to inertial coordinate system.

$(\overline{R}_k)_o$ = position vector of k^{th} body reference point relative to the body coordinate system.

$\overline{r}_{ki}$ = position vector of point i relative to the strut or k^{th} body coordinate system.

k = Subscript defining a specific body. The airplane minus its gears is the 0^{th} body (k = 0). A typical landing gear is the k^{th} body (k = 1, 2, 3...K).

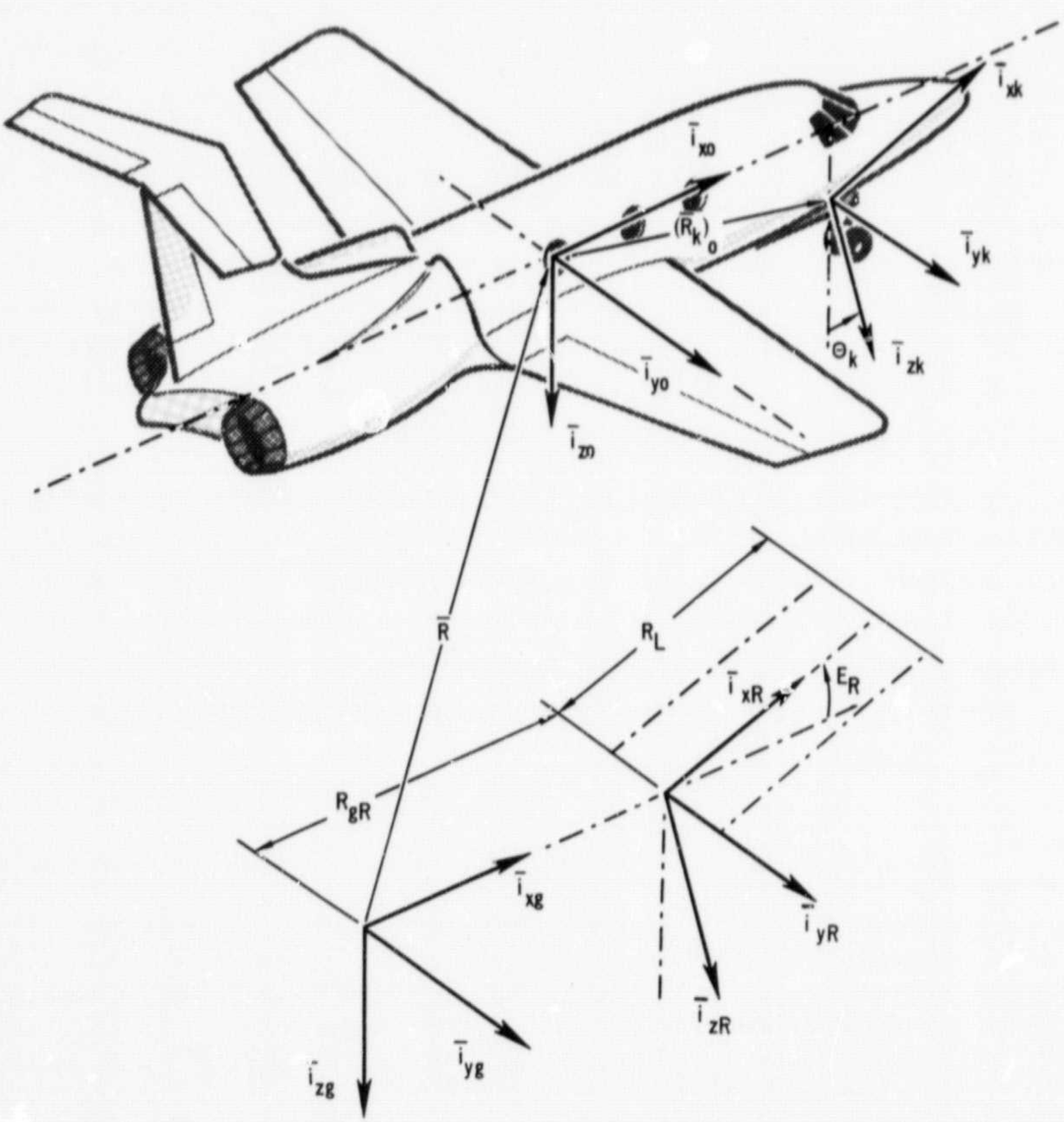


FIGURE 2–2 COORDINATE SYSTEMS

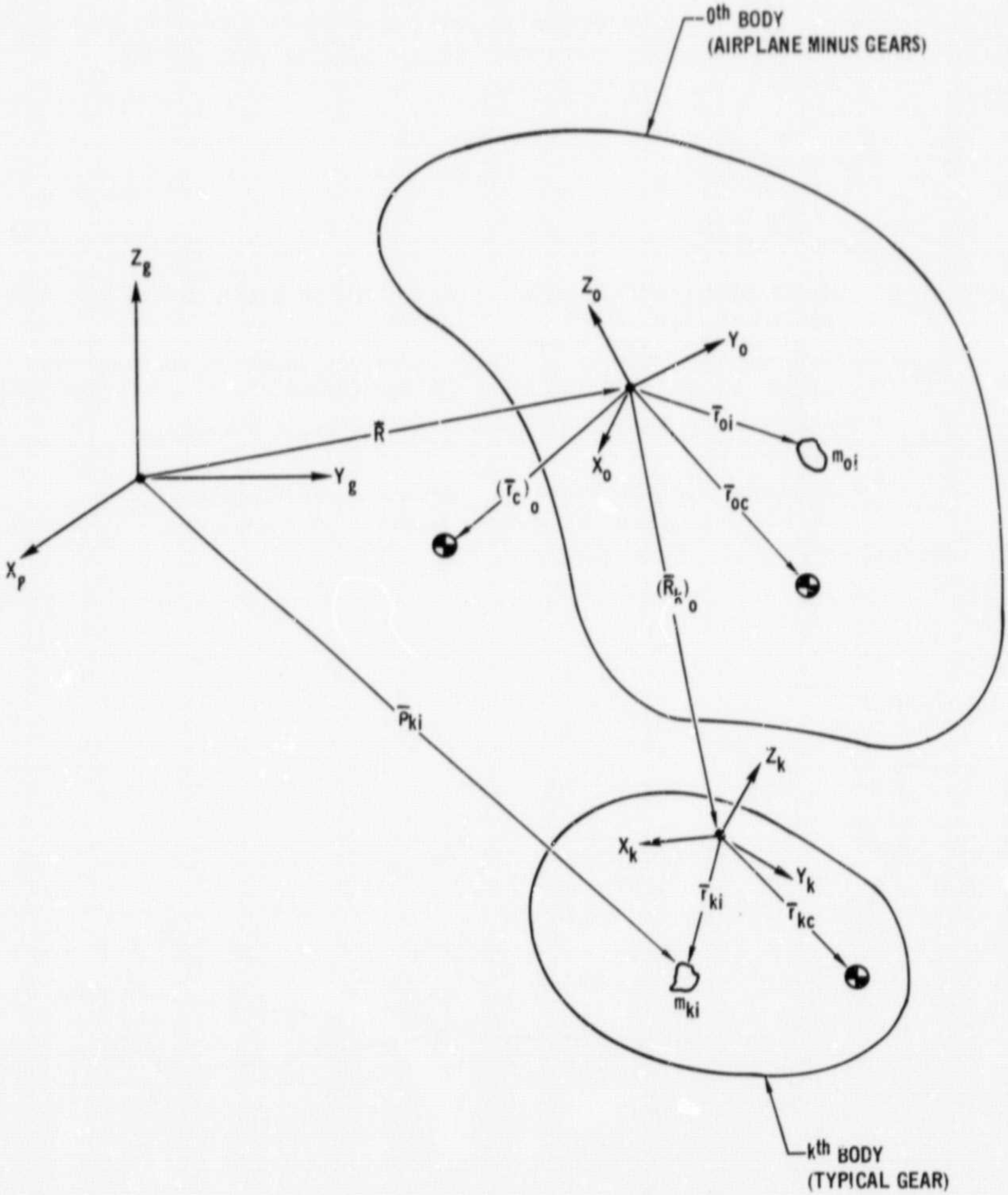


FIGURE 2–3 POSITION VECTORS

It is assumed that the position vectors $(\overline{R}_k)_o$ and $\overline{r}_{ki}$ used to locate point i in equation (2-1) can be separated into terms which vary with time and terms which remain constant with time. Thus, these vectors may be written

$$(\overline{R}_k)_o = \overline{R}_{ks} + \overline{R}_{ke}(t)$$

$$\overline{r}_{ki} = \overline{r}_{kis} + \overline{r}_{kie}(t) \tag{2-2}$$

where $\overline{R}_{ks}$ = undeformed position of k^{th} body reference point in the body coordinate system

$\overline{R}_{ke}(t)$ = deformed position of k^{th} body reference point in the body coordinate system measured from the undeformed position of that point

$\overline{r}_{kis}$ = undeformed position of point i in the strut coordinate system.

$\overline{r}_{kie}(t)$ = deformed position of point i in the strut coordinate system measured from the undeformed position of that point

These position vectors are shown in Figure 2.4.

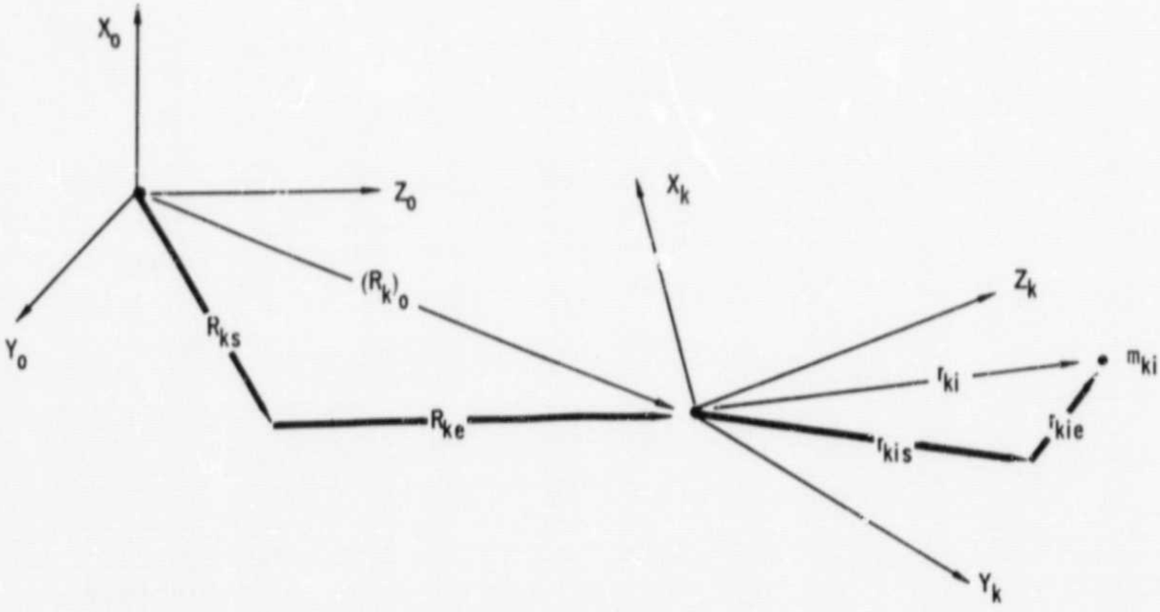


FIGURE 2-4 POSITION OF POINT i RELATIVE TO BODY COORDINATE SYSTEM

Since it is assumed that the elastic deformation of the airplane main body can be represented by the superposition of a limited number of vibratory modes, the terms of $(\bar{R}_k)_o$ and $\bar{r}_{ki}$ that vary with time may be written as

$$\bar{R}_{ke}(t) = \sum_{n=1}^{N} \bar{\phi}_n^k \, q_n(t)$$

and for $k = 0$ (2-3)

$$\bar{r}_{oie}(t) = \sum_{n=1}^{N} \bar{\phi}_n^i \, q_n(t)$$

where N = number of vibratory modes included

$\bar{\phi}_n^i, \bar{\phi}_n^k$ = magnitude of n^{th} elastic mode at points i and k respectively. (k used as a superscript refers to attachment point of k^{th} body, $k = 1, 2, 3 \ldots K$)

q_n = generalized coordinate associated with the n^{th} mode. These are a function of time.

Combining the above equations gives the final form for the location of a point i (Figure 2-3) as

$$\bar{\rho}_{ki} = \bar{R} + \bar{R}_{ks} + \sum_{n=1}^{N} \bar{\phi}_n^k \, q_n(t) + \bar{r}_{kis} + \sum_{n=1}^{N} \bar{\phi}_n^i \, q_n(t) \qquad (2\text{-}4)$$

The kinetic energy of the K+1 bodies, T, is obtained by summing the kinetic energy of all points i in the system having mass m_{ki}

$$T = \frac{1}{2} \sum_{k=0}^{K} \sum_{i=1}^{I} m_{ki} \, \dot{\bar{\rho}}_{ki} \cdot \dot{\bar{\rho}}_{ki} \qquad (2\text{-}5)$$

where K = total number of landing gears

I = total number of mass points

$\dot{\bar{\rho}}_{ki}$ = time derivative of the displacement vector

The potential energy, U, due to the strain energy of the main or 0^{th} body is

$$U = \frac{1}{2} \sum_{n=1}^{N} \omega_n^{\;2} \; m_n \; q_n^2 \tag{2-6}$$

where ω_n = the natural frequency of the n^{th} free-free mode

m_n = generalized mass of n^{th} elastic mode

The airplane's equations of motion were obtained by using the Lagrangian equations and the energy expressions of (2-5) and (2-6). The Lagrangian equations may be written

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_p}\right) - \frac{\partial T}{\partial q_p} + \frac{\partial U}{\partial q_p} = Q_p \tag{2-7}$$

where q_p, $\dot{q}_p$ = p^{th} generalized coordinate and generalized velocity respectively

Q_p = generalized force or moment in p^{th} mode

The resulting rigid body translational, rigid body rotational, and the flexible body equations of motion are given in Equations (2-8), (2-9), and (2-10) respectively.

Certain simplifying assumptions were made in the equations of motion. These assumptions were consistent with those made in the existing TOLA formulation. In the rigid body equations, the Coriolis acceleration terms for the landing gear struts were neglected mainly because the rotational velocities of the aircraft are small at landing relative to strut acceleration terms retained. Also, in the rigid body rotational equations, terms involving the variation of the inertia or inertia derivative tensors resulting from changing position vectors of the strut masses are small. The changes in the

RIGID BODY TRANSLATIONAL EQUATIONS

$$\bar{F}_T = M_T \ddot{\bar{R}} - \sum_{k=1}^{K} m_k \ddot{S}_k \, [A_{k31} \, \bar{i}_{xo} + A_{k33} \, \bar{i}_{zo}] + \sum_{n=1}^{N} \sum_{k=1}^{K} m_k \ddot{q}_n \, [\phi_{xn}^k \, \bar{i}_{xo}$$

$$+ \phi_{yn}^k \, \bar{i}_{yo} + \phi_{zn}^k \, \bar{i}_{zo}] \qquad (2\text{-}8)$$

RIGID BODY ROTATIONAL EQUATIONS

$$\bar{M}_o = \bar{\bar{I}} \cdot \dot{\bar{\omega}} + \bar{\omega} \times [\bar{\bar{I}} \cdot \bar{\omega}] + \sum_{k=1}^{K} m_k \ddot{S}_k \, [- A_{k11} \, R_{ksy} \, \bar{i}_{xo} + (A_{k13} \, R_{ksz}$$

$$+ A_{k11} \, R_{ksx}) \, \bar{i}_{yo} - A_{k13} \, R_{ksy} \, \bar{i}_{zo}]$$

$$+ \sum_{n=1}^{N} \sum_{k=1}^{K} m_k \ddot{q}_n \, \Big\{ [R_{ksy} \, \phi_{zn}^k - (R_{ksz} + A_{k11} \, (r_{Fk} - S_{kc})) \, \phi_{yn}^k] \, \bar{i}_{xo} \qquad (2\text{-}9)$$

$$+ [(R_{ksz} + A_{k11} \, (r_{Fk} - S_{kc})) \, \phi_{xn}^k - (R_{ksx} - A_{k13} \, (r_{Fk} - S_{kc})) \, \phi_{zn}^k] \, \bar{i}_{yo}$$

$$+ [(R_{ksx} - A_{k13} \, (r_{Fk} - S_{kc})) \, \phi_{yn}^k - R_{ksy} \, \phi_{xn}^k] \, \bar{i}_{zo} \Big\}$$

Symbols are defined on the following page

RIGID BODY EQUATIONS OF MOTION

F_T = total applied force acting on K+1 bodies

M_T = total mass of K+1 bodies

m_k = mass of the k^{th} strut

$\ddot{S}_k$ = acceleration of k^{th} strut mass

A_{k11}, A_{k13}, A_{k31}, A_{k33} = direction cosines relating strut coordinate system to 0^{th} body coordinate system

$\bar{M}_o$ = total moment of all applied forces on K+1 bodies about 0^{th} body center of mass

$\bar{\bar{I}}$ = inertia tensor for the K+1 bodies about 0^{th} body center of mass

$\bar{\omega}$ = angular velocity of 0^{th} body in body coordinate system

R_{ksx}, R_{ksy}, R_{ksz} = X, Y, Z components of strut attach point position vector

r_{Fk} = distance from strut attach point to extended position of axle

S_{kc} = distance from strut axle to strut center of mass

FLEXIBLE AIRFRAME EQUATIONS OF MOTION

FOR $S = 1, 2, 3, \ldots N$

$$m_s \ddot{q}_s + \omega_s^2 m_s q_s + \sum_{n=1}^{N} \sum_{k=1}^{K} m_k [\phi_{xs}^k \phi_{xn}^k + \phi_{ys}^k \phi_{yn}^k + \phi_{zs}^k \phi_{zn}] \ddot{q}_n$$

$$+ \sum_{k=1}^{K} m_k [(\ddot{X} - A_{k31} \ddot{S}_k) \phi_{xs}^k + \ddot{Y} \phi_{ys}^k + (\ddot{Z} - A_{k33} \ddot{S}_k) \phi_{zs}^k]$$

$$+ \dot{\omega}_x \sum_{k=1}^{K} m_k [R_{ksy} \phi_{zs}^k - (R_{ksz} + A_{k11} (r_{Fk} - S_{kc})) \phi_{ys}^k] \tag{2-10}$$

$$+ \dot{\omega}_y \sum_{k=1}^{K} m_k [(R_{ksz} + A_{k11} (r_{Fk} - S_{kc})) \phi_{xs}^k - (R_{ksx} - A_{k13} (r_{Fk} - S_{kc})) \phi_{zs}^k]$$

$$+ \dot{\omega}_z \sum_{k=1}^{K} m_k [(R_{ksx} - A_{k13} (r_{Fk} - S_{kc})) \phi_{ys}^k - R_{ksy} \phi_{xs}^k] = Q_s$$

$Q_s = Q_s^T + Q_s^{DC} + Q_s^A$ = Generalized forces associated with s th mode due to engine thrust, drag chute deployment and aerodynamic pressure respectively

Symbols are defined on the following page

FLEXIBLE AIRFRAME EQUATIONS OF MOTION

m_s, q_s, ω_s = Generalized mass, generalized coordinate and natural frequency respectively associated with the s th elastic free-free mode

$$Q_s^T = \phi_{xs}^{T_1} T_{x1} + \phi_{xs}^{T_2} T_{x2} + \phi_{xs}^{T_3} T_{x3} + \phi_{xs}^{T_4} T_{x4}$$

T_{x1} through T_{x4} = Engine thrust forces

$\phi_{xs}^{T_1}$ through $\phi_{xs}^{T_4}$ = Magnitude of s th mode shape at point force application

$$Q_s^{DC} = \phi_{xs}^{DC} F_{cx} + \phi_{ys}^{DC} F_{cy} + \phi_{zs}^{DC} F_{cz}$$

F_{cx}, F_{cy}, F_{cz} = Forces due to deployment of drag chute

ϕ_{xs}^{DC}, ϕ_{ys}^{DC}, ϕ_{zs}^{DC} = Magnitude of s th mode shape at point of force application

$$Q_s^A = -B_s^1 \phi_{xs}^A a + B_s^2 \phi_{ys}^A y - B_s^3 \phi_{zs}^A n_f + B_s^4 \theta_{xs}^A l + B_s^5 \theta_{ys}^A m + B_s^6 \theta_{zs}^A n$$

a, y, n_f = Aerodynamic forces; l, m, n = aerodynamic moments

ϕ_{xs}^A, ϕ_{ys}^A, ϕ_{zs}^A = Magnitude of s th mode shape at point of force application

θ_{xs}^A, θ_{ys}^A, θ_{zs}^A = Slope of s th mode shape at point of moment application

B_s^1 through B_s^6 = Participation factors for s th mode

gear positions are very small compared to the overall airplane dimensions, therefore the variation of the inertia terms due to gear position changes are small and were neglected. The flexible airframe equations of motion were also simplified by neglecting terms analogous to those neglected in the rigid body equations. Coriolis and centripetal type accelerations of the strut masses were considered small, again mainly because the rotational velocities of the aircraft are small relative to the strut acceleration terms retained. Rigid body "tangential" type acceleration terms that vary with changing position vectors were neglected; however, all others were retained.

Reference 3 gives a more thorough derivation of the total aircraft equations of motion. In addition, topics such as the determination and implementation of elastic airframe effects on strut equations of motion are also discussed. The reader will find additional information on these and other aspects of the flexible body option version of TOLA presented in Reference 3 that will aid his understanding of the program operation.

3. PROGRAM DESCRIPTION

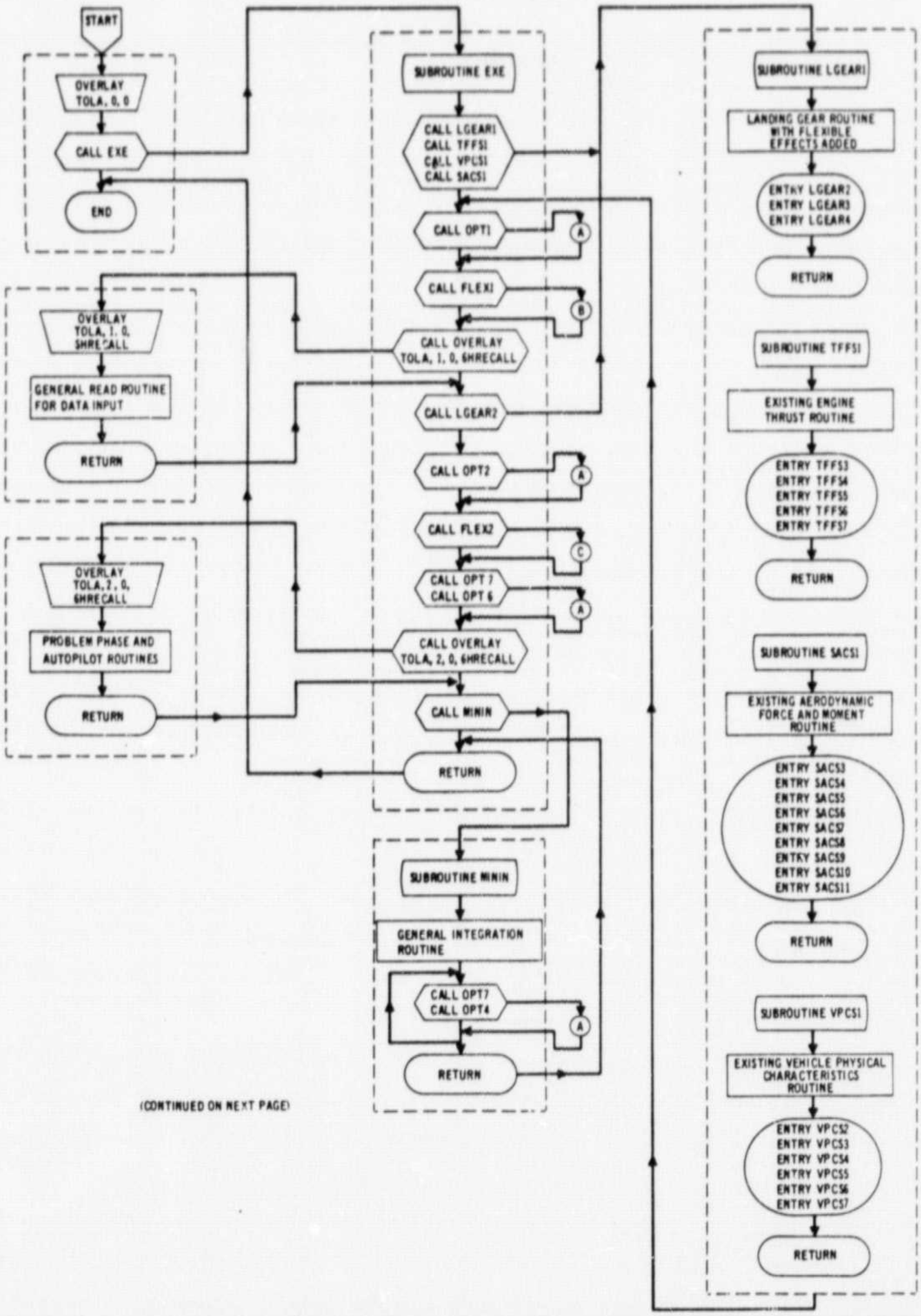
3.1 Program Structure - The equations describing the motion of points on a flexible airframe are formulated and solved in subroutine FLEX1. This subroutine can determine the response at twenty points on an airframe whose stiffness characteristics are defined by the modal deflections and natural frequencies for a maximum of twenty modes. A matrix approach was used to simultaneously formulate and solve all flexible body equations of motion at a particular point in time.

The basic structure of TOLA has been retained. It still consists of three OVERLAY segments. OVERLAY (TOLA, 0, 0) consists of the executive subprogram TOLA. Through its call to subroutine EXE, it calls the other two overlays and controls the execution of the complete program. TOLAN1 is the executive subprogram in OVERLAY (TOLA, 1, 0). It reads the input data and checks to determine if the proper amount of input data has been supplied. TOLAN2 is the executive subprogram in OVERLAY (TOLA, 2, 0). It provides the autopilot and phasing functions. A diagram showing the general structure of the revised TOLA program is given in Figure 3-1. This diagram is not intended to be a comprehensive programming chart, but shows the general flow of the flexible body option logic. A listing of the TOLA computer program is given in Appendix A.

Subroutine FLEX1 is located in OVERLAY (TOLA, 0, 0). The format used in structuring this subroutine closely follows that of existing subroutines so as to make it compatible with the logic used in the TOLA program. This structure enables calls to entries in FLEX1 to be placed in the EXE and OPT1 subroutines at those points where calls are made to other subprograms that perform similar functions.

All input data associated with the flexible body option are stored in COMMON/DIRCOM/. Those data calculated within the subroutine and passed to other parts of the program are stored in COMMON/FLXOP/. This use of common simplifies the communication between subroutines.

3.2 Program Operation - An understanding of program development in itself will not result in smooth program operation. Successful operation of the TOLA computer program with the flexible body option depends on proper input of all required data. Much of these data are associated with the existing rigid body version of TOLA. It is not the purpose of this document to define



(CONTINUED ON NEXT PAGE)

FIGURE 3-1 FLOW DIAGRAM FOR TOLA COMPUTER PORGRAM FLEXIBLE BODY OPTION

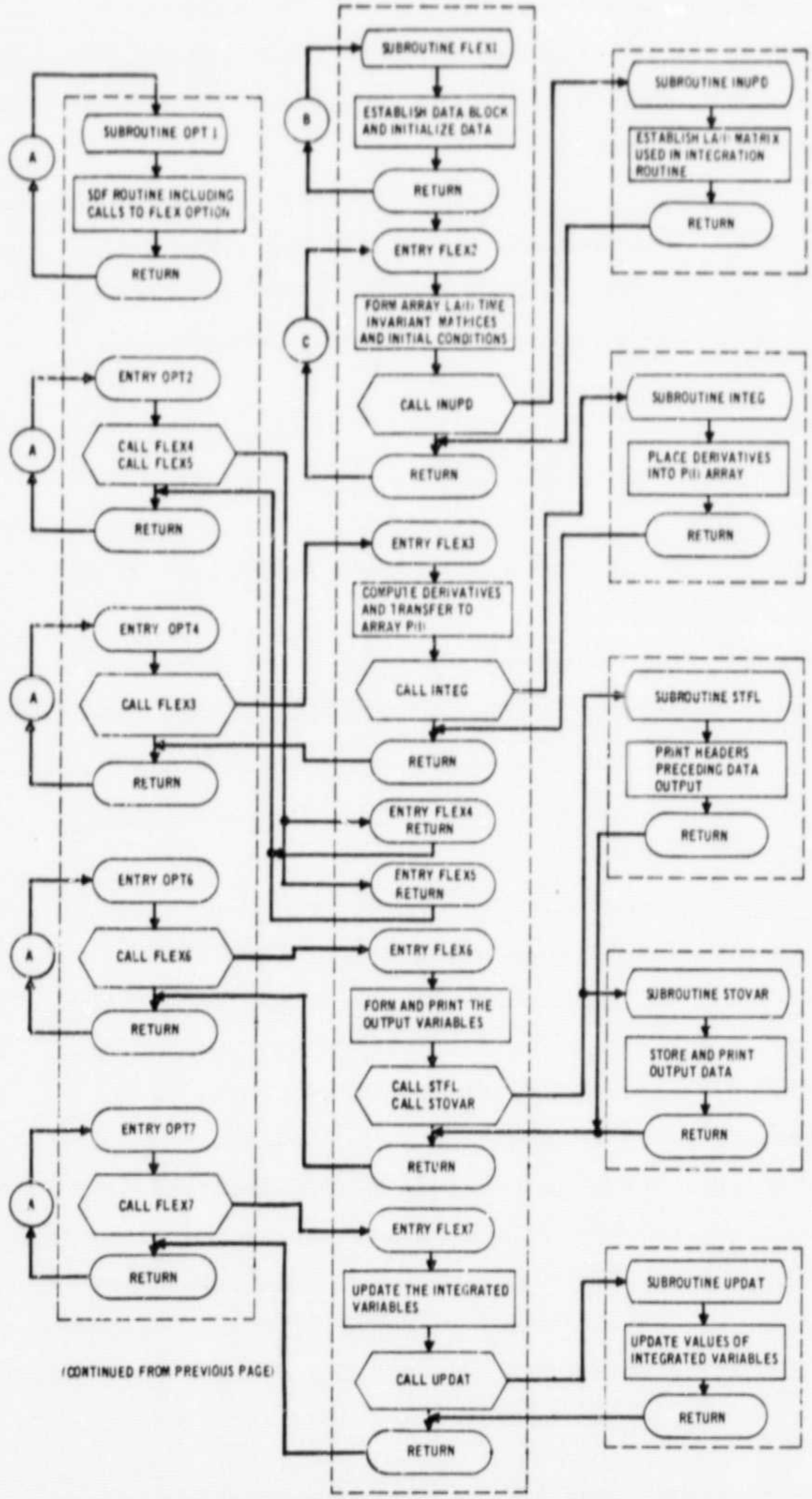


FIGURE 3–1 FLOW DIAGRAM FOR TOLA COMPUTER PROGRAM FLEXIBLE BODY OPTION (Continued)

these data, for this is done in Reference 1. The sections that follow will describe the general input data format, define all possible data associated with the flexible body option and the manner in which it is to be input, and discuss the resulting output.

3.2.1 General Input Data Format - The input data associated with the flexible body option are read into the program by the existing input routine (READ). As a result, it must follow the same general format as the rigid body data. The input must follow the following format:

Card Columns	1-6	7	8-10	11	12-66	67-72	73-80
Field Number	I	II	III	IV	V	VI	VII

Field number I contains the alphanumeric variable name of the data contained in Field V. Example:

```
STATEMENT NUMBER | FORTRAN STATEMENT
INDFLX     1
```

The program will assign a value of "1" to the variable INDFLX.

Field number II is not used.

Field number III is used to define the type of data in Field V by means of the words DEC, BCD, INT, TRA or a blank. DEC and blank are equivalent and denote that data which follow are in decimal form. The work ØCT specifies that the data in Field V are to be interpreted as octal numbers. The word BCD specifies that N binary coded, six character words (N punched in column 12) are to be loaded. These decimal words begin in column 13. The largest number of six character words that can be loaded from one card is nine. The user should make sure that BCD data does not get punched into Field VI, or an input error will result. When the word INT is used, it is assumed that all numbers in Field V will be loaded as integers. If only one integer is punched per card, the INT may be omitted. The word TRA denotes to the read routine that all data have been input and to return control to the calling program. Example:

```
STATEMENT NUMBER | FORTRAN STATEMENT
REM    BCD 4FLEXIBLE BODY OPTION
FREQ   DEC 2.0,5.67,200.0
       TRA
```

On the first card, the 4 in column 12 indicates that four words follow; each word is six characters long. This is a remarks card which helps identify the data that follows. The second card defines a variable array FREQ(I) whose elements are 2.0, 5.67 and 200.0 respectively. The third card denotes the end of the data.

Field number IV is not used.

Field number V contains the data input to the program. DEC, ØCT and INT numbers must be left adjusted; that is, data must always start in column 12. All numbers must be separated by a comma and the field terminates with the first blank. BCD information begins in column 13. Because the field ends with a blank, the user may add any comments in the remainder of the field.

Field number VI specifies the initial array location or subscript of the data in Field V. If this field is blank, an initial subscript of 1 is implied. The subscript may appear anywhere in the field. Example:

```
STATEMENT NUMBER                         FORTRAN STATEMENT
SXMOD      0.075,0.893,0.942
SXMOD      1.0,0.331                                              4
```

In the example, SXMOD is an array containing five elements. The first element is 0.075 while the fourth element is 1.0 as denoted by the 4 in column 67.

Field number VII is not used as far as the input routine is concerned.

3.2.2 Input Data for the Flexible Body Option - The revised TOLA computer program can be run considering either a rigid or a flexible airframe. In either case, all of the data defined in Reference 1 for a particular case must be input. The additional data required by the flexible body option of the TOLA computer program are defined in Figure 3-2 and discussed below.

To use TOLA when the airframe is considered rigid, the indicator INDFLX must be equal to zero. No other additional input is required.

To use TOLA when the airframe is considered flexible, the indicator INDFLX must be set equal to one and additional data describing the airframe elasticity are required. The number of free-free normal modes, NMODE, representing the airframe flexibility must be specified. A maximum of twenty modes

INPUT INDICATOR/ VARIABLE	COORDINATE SYSTEM(1)	INDICATOR OR VARIABLE DEFINITION
INDFLX		INDICATOR DEFINING OPTION INDFLX = 0 - RIGID AIRFRAME INDFLX = 1 - FLEXIBLE AIRFRAME
NMODE		INDICATES NUMBER OF MODES INPUT (20 MAXIMUM)
GMASS1(I)		GENERALIZED MASS OF I TH ELASTIC MODE
GFREQ(I)		FREQUENCY OF I TH MODE (RADIANS/SEC)
SXMOD(I)	BCS	X MODE SHAPE FOR STRUT ATTACHMENT POINTS IN I TH MODE
SYMOD(I)	BCS	Y MODE SHAPE FOR STRUT ATTACHMENT POINTS IN I TH MODE
SZMOD(I)	BCS	Z MODE SHAPE FOR STRUT ATTACHMENT POINTS IN I TH MODE
TXMOD(I)	BCS	X MODE SHAPE FOR ENGINE THRUST APPLICATION ATTACHMENT POINTS IN I TH MODE
ARMODE(I)	BCS	X,Y,Z,θ_x,θ_y,θ_z COMPONENTS OF MODE SHAPE FOR AERODYNAMIC FORCE AND MOMENT REFERENCE LOCATION IN I TH MODE
PF(I)		PARTICIPATION FACTORS OF GENERALIZED AERODYNAMIC FORCES AND MOMENTS IN I TH MODE
SKC(K)	SCS	DISTANCE BETWEEN K TH STRUT AXLE AND STRUT CENTER OF MASS
DCMODE(I)	BCS	X,Y,Z COMPONENTS OF MODE SHAPE FOR DRAG CHUTE ATTACHMENT POINT IN I TH MODE
NPTS		NUMBER OF POINTS ON THE AIRFRAME AT WHICH OUTPUT IS REQUESTED (20 MAXIMUM)
ROIS(J)	BCS	ROISX, ROISY, ROISZ COMPONENTS OF POSITION VECTOR FOR J TH POINT ON FLEXIBLE AIRFRAME AT WHICH OUTPUT IS REQUESTED
OUTMOD(I)	BCS	X,Y,Z COMPONENTS OF I TH MODE SHAPE FOR POINTS ON FLEXIBLE AIRFRAME AT WHICH OUTPUT IS REQUESTED.
IFLX(I)		INDICATES THE POINTS WHOSE DATA WILL BE STORED ON TAPE

(1) NOTE: BCS = BODY COORDINATE SYSTEM SCS = STRUT COORDINATE SYSTEM

FIGURE 3-2 INPUT DATA FOR TOLA FLEXIBLE BODY OPTION

can be used. The natural frequency, GFREQ(I), and generalized mass, GMASS1(I), for each mode must be input. The mode shape magnitudes, SXMOD(I), SYMOD(Y), SZMOD(Z), for each mode at the strut attachment locations should be input. While the program will run without these input data, it will use a value of zero for the mode shape magnitudes. Similarly, the mode shape data at the point of application of the engine thrust, TXMOD(I), and aerodynamic forces and moments, ARMODE(I), should be input for solution of any practical takeoff/landing problem.

The aerodynamic loads are treated as concentrated loads in TOLA in the form of total aerodynamic forces and moments acting at a selected reference point on the airframe. To obtain realistic flexible body response, weighting effects or participation factors, PF(I), of the aerodynamic loads on the response of each normal mode are required. If aerodynamic effects on the flexible airframe response are desired, the user must input the proper values for the PF(I) array. If the PF(I) are not input, the default values for this array are zero and all effects of the aerodynamic loads on modal response are assumed to be small and therefore neglected. A general approach for calculating the aerodynamic weighting effects is suggested in Appendix B.

All of the arrays used in the flexible body option are in vector form (one dimensional), and the subroutine expects the modal data to appear in a particular order on the data input cards. The order required by the subroutime is best shown through an example. If the aircraft simulated in the program has three landing gear struts and two engines, and four vibratory modes are selected to represent airframe flexibility, then the typical strut and engine attachment point modal data would appear as follows:

Strut 1 Attach Point

Mode	X	Y	Z
1	-.41	.68	-.14
2	.38	.03	.79
3	-.40	-.08	.03
4	-.25	.16	.17

Strut 2 Attach Point

Mode	X	Y	Z
1	-.83	.69	-.33
2	.48	.06	.86
3	-.83	-.07	.05
4	-.47	.17	.38

Strut 3 Attach Point

Mode	X	Y	Z
1	-1.00	.69	-.41
2	.52	.07	.89
3	-1.00	-.08	.05
4	-.55	.17	.49

Engine 1 Attach Point

Mode	X
1	-.64
2	.29
3	-.59
4	-.32

Engine 2 Attach Point

Mode	X
1	-.68
2	.36
3	-.65
4	-.37

Write each component of the strut attachment point mode shape as though in an array dimensioned number-of-struts x number-of-modes.

$$[SXMOD] = \begin{bmatrix} -.41 & .38 & -.40 & -.25 \\ -.83 & .48 & -.83 & -.47 \\ -1.0 & .52 & -1.0 & -.55 \end{bmatrix}$$

$$[SYMOD] = \begin{bmatrix} .68 & .03 & -.08 & .16 \\ .69 & .06 & -.07 & .17 \\ .69 & .07 & -.08 & .17 \end{bmatrix}$$

$$[SZMOD] = \begin{bmatrix} -.14 & .79 & .03 & .17 \\ -.33 & .86 & .05 & .38 \\ -.41 & .89 & .05 & .49 \end{bmatrix}$$

The engine attachment point modal data should be written in an array dimensioned number-of-engines x number-of-modes:

$$[TXMOD] = \begin{bmatrix} -.64 & .29 & -.59 & -.32 \\ -.68 & .36 & -.65 & -.37 \end{bmatrix}$$

Transform each array into vector form by reading the matrix elements by columns. Thus the order in which the data is read would be as follows:

SXMOD =	SYMOD =	SZMOD =	TXMOD =
-.41	.68	-.14	-.64
-.83	.69	-.33	-.68
-1.0	.69	-.41	.29
.38	.03	.79	.36
.48	.06	.86	-.59
.52	.07	.89	-.65
-.40	-.08	.03	-.32
-.83	-.07	.05	-.37
-1.0	-.08	.05	
-.25	.16	.17	
-.47	.17	.38	
-.55	.17	.49	

These data would then be placed on cards according to the format described in the previous section.

Typical aerodynamic data and participation factors would appear as follows:

Aerodynamic Data

Mode	X	Y	Z	θ_X	θ_Y	θ_Z
1	.16	.20	-.89	.0004	.0045	.0009
2	.20	.15	-.96	.0002	.0053	-.0003
3	.12	.08	-.65	-.0001	.0038	.0010
4	.05	.09	-.59	.0003	.0036	-.0007

Participation Factors

Mode	X	Y	Z	θ_X	θ_Y	θ_Z
1	.30	.36	-1.87	.15	-.93	.20
2	.46	.39	2.15	.12	-1.15	-.09
3	.28	.21	-1.13	.04	-.83	.21
4	.16	.24	.97	.08	-.79	-.17

Again, write the data in an array where each column represents the data for a single mode; then read the matrix columnwise.

$$
ARMODE = \begin{bmatrix} .16 & .20 & .12 & .05 \\ .20 & .15 & .08 & .09 \\ -.89 & -.96 & -.65 & -.59 \\ .0004 & .0002 & -.0001 & .0003 \\ .0045 & .0053 & .0038 & .0036 \\ .0009 & -.0003 & .0010 & -.0007 \end{bmatrix} \qquad PF = \begin{bmatrix} .30 & .46 & .28 & .16 \\ .36 & .39 & .21 & .24 \\ -1.87 & 2.15 & -1.13 & .97 \\ .15 & .12 & .04 & .08 \\ -.93 & -1.15 & -.83 & -.79 \\ .20 & -.09 & .21 & -.17 \end{bmatrix}
$$

ARMODE = .16	PF = .30
.20	.36
-.89	1.87
.0004	.15
.0045	-.93
.0009	.20
.20	.46
.15	.39
.	.
.	.
.	.

SKC(K) and DCMODE(I) are optional data. The program sets these quantities to zero if not input. SKC(K) is the distance from the tire axle to the strut center of mass measured along the strut stroke. It is approximately equal to zero for many landing gears. The order in which the struts are numbered in SKC(K) must be consistent with the order implied by the strut attachment point mode shapes. If the third row in the modal data represents the motion of the third strut attachment point, the third element in array SKC(K) must be the described distance for that strut.

DCMODE(I) defines the mode shape at the drag chute attach point and is necessary only when the aircraft has a drag chute. These modal data are input in an order similar to that for the aerodynamic data. If written in a two-dimensional array, each column of data represents the X, Y, Z components of the mode shape for a given mode.

The remaining input data indicated in Figure 3-2 are necessary to obtain flexible body response output on the airframe. NPTS indicates the number of points at which output is requested. ROIS(J) are the X, Y, Z components of

the position vector defining the location of the points relative to the airframe or 0th body coordinate system. OUTMOD(I) are the components of each mode shape at those points for which output is specified. As before, the order of the data is significant and an example will best show this. If output is desired at two points on the aircraft, typical data would appear as below:

Point 1

Location	Mode	1	2	3	4
X = 2.71	X	.09	.11	-.05	.02
Y = 17.63	Y	.02	.01	-.03	.01
Z = 1.52	Z	.95	-.98	.87	-.77

Point 2

Location	Mode	1	2	3	4
X = 2.71	X	.07	.10	-.04	.01
Y = 8.42	Y	.01	-.02	-.03	-.02
Z = 1.25	Z	.43	-.39	.41	-.29

The matrix ROIS(J) should be written as though dimensioned 3(X,Y,Z) x NPTS and then read by columns. For the above example, ROIS(J) would appear as follows:

ROIS(J) = 2.71, 17,63, 1.52, 2.71, 8.42, 1.25

Each column of OUTMOD(I) should contain all of the modal data for a given point. The X component of data for all modes should procede the Y component, with the Y component preceding the Z. Written in two dimensional form for the example being considered, OUTMOD(I) would be given as:

$$[\text{OUTMOD}] = \left[\begin{array}{rr} .09 & .07 \\ .11 & .10 \\ -.05 & -.04 \\ .02 & .01 \\ .02 & .01 \\ .01 & -.02 \\ -.03 & -.03 \\ .01 & -.02 \end{array}\right. \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{X Component} \\ \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{Y Component} \end{array}$$

$$\left.\begin{bmatrix} .95 & .43 \\ -.98 & -.39 \\ .87 & .41 \\ -.77 & -.29 \end{bmatrix}\right\} \text{ Z Component}$$

This matrix should be transformed to vector form by reading the elements by column.

The array IFLX(I) dictates which flexible body response data will be saved on tape and used as input for a plot routine. Subroutine FLEX can formulate and print the flexible body response at up to twenty points on the aircraft. IFLX(I) enables the user to select from these points, those whose data will be plotted. Each output point is assigned a number by the order in which their modal data appears in array ØUTMØD(I). An element in IFLX whose value is one will cause the data for that respective point to be placed on tape. For example, if response data is formulated at five points on the aircraft, IFLX = (1, 0, 1, 1, 0) will cause all flexible body data associated with point numbers one, three and four to be saved on tape. Other indicators required to store TOLA output data are discussed in Section 3.2.5. If no flexible body response data is desired, NPTS, RØIS, ØUTMØD and IFLX need not be input.

There is no specific system of units associated with the input information, except for the modal frequencies which must be expressed in radians/unit time. All other parameters may be expressed in any consistent set of units, either English or Metric (inches or centimeters, pounds or dynes). The units selected must, of course, be consistent with the rigid body set used (see Reference 1).

3.2.3 <u>Output Data from the Flexible Body Option</u> - The data that can be output from subroutine FLEX1 consists of the flexible body components of the inertial accelerations, velocity and displacement in each of the three body coordinate directions and the total inertial acceleration and velocity in each coordinate direction. The output variable names used in FLEX1 and their definition are given in Figure 3-3.

3.2.4 <u>Staging the Flexible Body Option into the Program</u> - If aircraft elasticity is desired in an analysis, the program will turn the flexible body option subroutine on at the same time the landing gear subroutine is staged

ALL QUANTITIES ARE IN BODY COORDINATES

POINT	-	DEFINES THE POINT NUMBER (1-20)
XD2F	-	X COMPONENT OF THE INERTIAL ACCELERATION DUE TO FLEXIBILITY
XD2T	-	X COMPONENT OF THE TOTAL INERTIAL ACCELERATION
YD2F	-	Y COMPONENT OF THE INERTIAL ACCELERATION DUE TO FLEXIBILITY
YD2T	-	Y COMPONENT OF THE TOTAL INERTIAL ACCELERATION
ZD2F	-	Z COMPONENT OF THE INERTIAL ACCELERATION DUE TO FLEXIBILITY
ZD2T	-	Z COMPONENT OF THE TOTAL INERTIAL ACCELERATION
XD1F	-	X COMPONENT OF THE INERTIAL VELOCITY DUE TO FLEXIBILITY
XD1T	-	X COMPONENT OF THE TOTAL INERTIAL VELOCITY
YD1F	-	Y COMPONENT OF THE INERTIAL VELOCITY DUE TO FLEXIBILITY
YD1T	-	Y COMPONENT OF THE TOTAL INERTIAL VELOCITY
ZD1F	-	Z COMPONENT OF THE INERTIAL VELOCITY DUE TO FLEXIBILITY
ZD1T	-	Z COMPONENT OF THE TOTAL INERTIAL VELOCITY
XDOF	-	X COMPONENT OF DISPLACEMENT DUE TO FLEXIBILITY
YDOF	-	Y COMPONENT OF DISPLACEMENT DUE TO FLEXIBILITY
ZDOF	-	Z COMPONENT OF DISPLACEMENT DUE TO FLEXIBILITY

FIGURE 3-3 OUTPUT VARIABLES USED IN FLEX1

into the program. This is done by testing the values of both the flexible body option indicator (INDFLX) and the landing gear indicator (INDLG). Both must be non-zero for flexibility effects to be included.

Initial generalized displacements are calculated based on values of variables at the time the option is staged into the program. These initial displacements are then used in subsequent calculation of the generalized accelerations.

3.2.5 Data Plotting Information - Major revisions were made to the subroutine that stores data for use by a separate plotting program. As now rewritten, subroutine SDFLGP first prints headers that identify the variable names of the data that follow. These headers are printed only once. Each call to the subroutine then writes to tape all data associated with a single time point. This eliminates the need to store the data in intermediate arrays.

The indicators described in Reference 1 and earlier in this report that control the logic in SDFLGP have not been modified. They are as follows:

IPLT = 1	denotes that data will be stored on tape for plotting
ISDF = 1	indicates rigid body data will be saved
ISTPL1 = 1 ISTPL2 = 1 ISTPL3 = 1 ISTPL4 = 1 ISTPL5 = 1	denotes that data for landing gears number 1 through 5 will be stored on tape
IFLX(I) = 1	indicates that flexible response data for the i th output point will be saved

A new plotting program was developed to be compatible with the data format of the revised TOLA subroutine SDFLGP. This program, entitled PLTDAT, is submitted as a separate job after a data tape has been generated by the TOLA program. The plot tape generated by PLTDAT can then be used by any off-line plotting device. A listing of the PLTDAT computer program is given in Appendix C.

The plot program was designed to permit the user a high degree of flexibility in the use of the program. Any variable stored for plotting may be chosen as the independent variable. Not only does this allow conventional time history plots to be made, but also such plots as the altitude of the

center of mass versus downrange position or strut acceleration versus strut stroke. A maximum of five dependent variables can be plotted on a single graph. This enables the user to make direct comparisons of several simultaneously displayed variables.

All graphs are scaled to an 8-1/2 x 11 inch page size; however, the actual plotting area depends on the number of dependent variables. The ordinate (dependent variable) axis is six inches in length. The length of the abcissa (independent variable) axis ranges from 9.4 inches for one dependent variable to 7.0 inches for five dependent variables.

The input data required by the plot program must follow a particular format. The first card contains an integer (format I6) that specifies the number of data points plotted per graph. The remaining cards control the number of plots, define graph titles and dictate the dependent and independent variables. This is accomplished by beginning each data card with a control identifier. These control identifiers are TITLE1, TITLE2, INDVAR, DEPVAR and PLOT. All identifiers must begin in column 1 with their associated data beginning in column 11. TITLE1 and TITLE2 permit a 40 character title and subtitle to be printed on the graph. If no titles are desired, these data are omitted. INDVAR defines the variable name on that card as the independent variable. DEPVAR defines a maximum of five variable names (format 5A10) as dependent variables. The word PLOT causes the graphs to be generated with the current titles, dependent and independent variables.

In order to uniquely identify all variables, a two digit numerical prefix is assigned to each repetative data name. For example, if TOLA subroutine SDFLGP has saved data for landing gears one, three, and five, PLTDAT will assign a 01 prefix to all data for gear one, a 02 prefix to all data for gear three, and a 03 prefix to all data for gear five. Thus the strut acceleration for gear five would be 03SD2 while the strut stroke for gear three would be 02S.

Figure 3-8 shows a sample of input data for the plotting program. TOLA subroutine SDFLGP has already saved flexible body data for three points on the aircraft and landing gear data for gear numbers one, three, four and five. The first line in the plotting data indicates that two hundred points will be plotted on each graph. The main title for the first plot is "TOLA TIME HISTORY" while the subtitle is "STRUT ACCELERATIONS". The independent variable is time.

The dependent variables are the strut accelerations corresponding to the first three gears for which data is stored, struts one, three, and four.

```
   200
TITLE1    TOLA TIME HISTORY
TITLE2    STRUT ACCELERATIONS
INDVAR    TIME
DEPVAR    01SD2     02SD2     03SD2
PLOT
TITLE2    STRUT DISPLACEMENTS
DEPVAR    01S       02S       03S
PLOT
TITLE2    FLEXIBLE ACCELERATION - PILOT STATION
DEPVAR    01ZD2F
PLOT
```

FIGURE 3-4 SAMPLE OF PLOTTING DATA

For the second graph, no main title is defined so the program uses the current title defined by TITLE1. The subtitle for this plot is "STRUT DISPLACEMENTS" as shown by the TITLE2 card. The independent variable has not been redefined and therefore is still time. The dependent variables are the strut displacements for gears one, three and four.

The third and final plot indicated by these data also uses the current main title but redefines the subtitle as "FLEXIBLE ACCELERATION-PILOT STATION". The independent variable is still time while the single dependent variable is the vertical acceleration at the first flexible body point.

The following standard CalComp plotting routines are called by PLTDAT:

PLOTS	PLOT	SCALE	NUMBER
LABEL	SYMBOL	AXIS	

3.2.6 Comments on Program Operation - In addition to a working knowledge of the program, the user is generally interested in the size of the program

and its operating cost. Using the existing OVERLAY structure, the TOLA computer program with the flexible body option has a core requirement of 67K octal words. Most of the space associated with the flexible body option itself is allocated to store the potentially large quantity of input modal data. Every effort was made during program development to streamline the option.

Program operating costs vary from one computer system to the next so it is not possible to develop a single cost formula. Several observations, however, can be made. The major factors that effect the cost of a TOLA run are the total number of integration steps and the number of variables that need to be integrated. The number of integration steps is dictated by the time length of the analysis and the integration step size. Although the user has little control over the step size chosen by the integration routine, he can directly input the time at which the program will terminate. Care should be taken to insure that the program does not continue to run beyond the points of interest. An approximation to determine the time of a flexible body run is given by the expression:

$$(\mathrm{TIME})_{\mathrm{FLEX}} \doteq (\mathrm{TIME})_{\mathrm{RIGID}} \cdot \left(\frac{(\mathrm{NO.\ OF\ INTEGRATED\ VARIABLES})_{\mathrm{FLEX}}}{(\mathrm{NO.\ OF\ INTEGRATED\ VARIABLES})_{\mathrm{RIGID}}} \right)^{1.4}$$

4. SUBROUTINE DESCRIPTIONS

A description of all subroutines added to the original version of the TOLA computer program is included in this section. Also, those existing equations that were modified to include flexibility effects are shown. The executive subroutine of the flexible body option is FLEX1. This routine contains calls to the other subroutines associated with the option, and as a group, formulates and solves the equations of motion for points on the flexible airframe.

4.1 FLEX1 - Subroutine FLEX1 formulates a maximum of twenty coupled, second order differential equations, and through matrix manipulation, solves the following set of simultaneous linear equations:

$$[A] \left\{\ddot{q}\right\} = \left\{B\right\} \tag{4-1}$$

The coefficient matrix [A] is a function of the generalized mass corresponding to each mode, the mode shape defined at each strut attach point, and the strut masses. The matrix B is a function of all external forces and terms considered as forces during each time increment. The independent variables $\left\{\ddot{q}\right\}$ are the second derivative of the generalized coordinates with respect to time.

The overall subroutine is divided into several entry points with each entry performing the function described below:

FLEX1 establishes data blocks and array sizes and initializes variables.

FLEX2 computes those components of matrices [A] and $\left\{B\right\}$ that are time invariant and determines the initial generalized displacements.

FLEX3 computes the time varying components of matrix $\left\{B\right\}$ and formulates the simultaneous linear equations as a function of $\ddot{q}$.

FLEX4 is null.

FLEX5 is null.

FLEX6 formulates and prints the output data generated by the flexible body option.

FLEX7 transfers time dependent variables to and from the integration routine.

4.2 DECOMP - Subroutine DECOMP is part of a package that solves the set of simultaneous equations defined in equation (4-1). DECOMP manipulates the coefficient matrix [A] into upper and lower triangular form which, combined with subroutine SOLVE, eliminates the need for a matrix inverse routine. Subroutine DECOMP is called from ENTRY FLEX2.

4.3 SOLVE - Subroutine SOLVE determines the values of $\{\ddot{q}\}$ based on the upper and lower triangular coefficient matrix calculated in DECOMP and the matrix $\{B\}$ determined in FLEX3. Subroutine SOLVE is called from ENTRY FLEX3.

4.4 SING - Subroutine SING prints error diagnostics and terminates the program if subroutine DECOMP cannot properly transform the coefficient matrix [A] into upper and lower triangular form. SING is called from DECOMP.

4.5 Matrix Subroutines - The subroutines that follow are standard CDC library matrix routines that are included with the option to make the program independent of any machine library shortcomings or limitations. All of these routines are called from ENTRY FLEX2 or ENTRY FLEX3.

ARAY - Subroutine ARAY converts a data array from single to double dimension. The CDC library title for this routine (ARRAY) was changed since there was an existing TOLA subroutine by that name.

CTIE - Subroutine CTIE adjoins two matrices by columns.

LOC - Subroutine LOC computes a vector subscript for an element in a matrix of specified storage mode.

GTPRD - Subroutine GTPRD gives the transpose product of two general matrices.

GMSUB - Subroutine GMSUB will subtract two general matrices.

GMADD - Subroutine GMADD adds two general matrices.

GMPRD - Subroutine GMPRD computes the product of two general matrices.

4.6 Modifications to Existing TOLA Equations - Existing TOLA equations were modified to include the effects of flexibility on landing gear displacements, velocities and accelerations. The three components of the strut attach point position vectors, R_{AX}, R_{AY}, R_{AZ}, were revised to include the generalized flexible displacement as follows:

$$\begin{Bmatrix} R_{AX} \\ R_{AY} \\ R_{AZ} \end{Bmatrix} = \begin{Bmatrix} R_{AX} \\ R_{AY} \\ R_{AZ} \end{Bmatrix} + \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \begin{Bmatrix} \sum_{n=1}^{N} \phi_{xn}^{k} \, qn \\ \sum_{n=1}^{N} \phi_{yn}^{k} \, qn \\ \sum_{n=1}^{N} \phi_{zn}^{k} \, qn \end{Bmatrix} \tag{4-2}$$

where l_i, m_i, and n_i transform the displacements from inertial axes to body axes systems.

The changes in the strut velocity equations resulted from introducing the generalized flexible velocity $\dot{q}$. As rewritten, the components of the strut attach point velocity vector are as follows:

$$\begin{aligned} R_{DX} &= R_{DX} + \left(q \sum_{n=1}^{N} \phi_{zn}^{k} - r \sum_{n=1}^{N} \phi_{yn}^{k} \right) q_n + \sum_{n=1}^{N} \phi_{xn}^{k} \dot{q}_n \\ R_{DY} &= R_{DY} + \left(r \sum_{n=1}^{N} \phi_{xn}^{k} - p \sum_{n=1}^{N} \phi_{zn}^{k} \right) q_n + \sum_{n=1}^{N} \phi_{yn}^{k} \dot{q}_n \\ R_{DZ} &= R_{DZ} + \left(p \sum_{n=1}^{N} \phi_{yn}^{k} - q \sum_{n=1}^{N} \phi_{xn}^{k} \right) q_n + \sum_{n=1}^{N} \phi_{zn}^{k} \dot{q}_n \end{aligned} \tag{4-3}$$

where p, q, and r are the rigid body roll, pitch and yaw rates respectively.

Similarly, the strut acceleration equations are modified by incorporating the generalized flexible acceleration $\ddot{q}$. The revised equation then takes on the form

$$\ddot{S} = \ddot{S} + a_{k31} \sum_{n=1}^{N} \phi_{xn}^{k} \ddot{q}_n + a_{k33} \sum_{n=1}^{N} \phi_{zn}^{k} \ddot{q}_n \tag{4-4}$$

All of these modifications to the landing gear equations were made in subroutine LGEA3C.

The forces and moments resulting from landing gear flexibility were incorporated into the rigid body equations as shown below.

$$F_X = F_X - \sum_{n=1}^{N} \sum_{k=1}^{K} m_k \phi_{xn}^{k} \ddot{q}_n$$

$$F_Y = F_Y - \sum_{n=1}^{N} \sum_{k=1}^{K} m_k \phi_{yn}^{k} \ddot{q}_n$$

$$F_Z = F_Z - \sum_{n=1}^{N} \sum_{k=1}^{K} m_k \phi_{zn}^{k} \ddot{q}_n$$

$$L = L - \sum_{n=1}^{N} \sum_{k=1}^{K} m_k \{R_{ksy} \phi_{zn}^{k} - [R_{ksz} + a_{k11} (r_{Fk} - S_{kc})] \phi_{yn}^{k}\} \ddot{q}_n \tag{4-5}$$

$$M = M - \sum_{n=1}^{N} \sum_{k=1}^{K} m_k \{[R_{ksz} + a_{k11} (r_{Fk} - S_{kc})] \phi_{xn}^{k} - [R_{ksx} - a_{k13} (r_{Fk} - S_{kc})] \phi_{zn}^{k} \} \ddot{q}_n$$

$$N = N - \sum_{n=1}^{N} \sum_{k=1}^{K} m_k \{-R_{ksy} \phi_{xn}^{k} + [R_{ksx} - a_{k13} (r_{Fk} - S_{kc})] \phi_{yn}^{k}\} \ddot{q}_n$$

These additions were made in subroutine LGEAR1.

5. FLOW LOGIC FOR FLEXIBLE BODY OPTION

Subroutine FLEX1 employs a matrix approach to formulate and solve the flexible body equations of motion. These equations are given as

$$
\begin{aligned}
&\left[m_s\right]\left\{\ddot{q}_s\right\} + \left[\omega_s^2 \; m_s\right]\left\{q_s\right\} \\
&+\left[\left[\phi^k_{xs}\right]^T \left[m_k\right]\left[\phi^k_{xn}\right] + \left[\phi^k_{ys}\right]^T \left[m_k\right]\left[\phi^k_{yn}\right] + \left[\phi^k_{zs}\right]^T \left[m_k\right]\left[\phi^k_{zn}\right]\right]\left\{\ddot{q}_n\right\} \\
&+\left[\left[\phi^k_{xs}\right]^T \,\Big|\, \left[\phi^k_{ys}\right]^T \,\Big|\, \left[\phi^k_{zs}\right]^T\right]\begin{bmatrix} \{m_k\} & 0 & 0 \\ 0 & \{m_k\} & 0 \\ 0 & 0 & \{m_k\} \end{bmatrix}\left[\begin{Bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{Bmatrix} - \begin{bmatrix} a_{k31} \\ 0 \\ a_{k33} \end{bmatrix}\left\{\ddot{S}_k\right\}\right] \\
&+\left(\left[\phi^k_{zs}\right]^T \left[R_{ksy}\right] - \left[\phi^k_{ys}\right]^T\left[\left[R_{ksz}\right] + \left[r_{Fk} - S_{kc}\right]\left[a_{k11}\right]\right]\right)\left\{\dot{\omega}_x m_k\right\} \\
&+\left(\left[\phi^k_{xs}\right]^T \left[\left[R_{ksz}\right] + \left[r_{Fk} - S_{kc}\right]\left[a_{k11}\right]\right] - \left[\phi^k_{zs}\right]^T\left[\left[R_{ksx}\right] - \left[r_{Fk} - S_{kc}\right]\right.\right. \\
&\qquad\left.\left.\left[a_{k13}\right]\right]\right)\left\{\dot{\omega}_y m_k\right\} \\
&+\left(\left[\phi^k_{ys}\right]^T \left[\left[R_{ksx}\right] - \left[r_{Fk} - S_{kc}\right]\left[a_{k13}\right]\right] - \left[\phi^k_{xs}\right]^T \left[R_{ksy}\right]\right)\left\{\dot{\omega}_z m_k\right\} = \left\{Q_s\right\} \qquad (5\text{-}1)
\end{aligned}
$$

Definition of these symbols with their program variable names are given in Figure 5-1.

Many of the terms in Equation (5-1) are constant and need be calculated only in the initial call to subroutine FLEX1 after the option has been staged into the program. Rewriting this equation in a shorthand matrix form that separates the constant terms from the time dependent variables yields

$$
\begin{aligned}
&[M]\ \{\ddot{q}\} + [K]\ \{q\} + [\mathrm{GFORC1}]\begin{Bmatrix} \ddot{X} - a_{k31}\ \ddot{S}_k \\ \ddot{Y} \\ \ddot{Z} - a_{k33}\ \ddot{S}_k \end{Bmatrix} + [\mathrm{GFORC2}]\ \left\{\dot{\omega}_x m_k\right\} \\
&+ [\mathrm{GFORC3}]\ \left\{\dot{\omega}_y m_k\right\} + [\mathrm{GFORC4}]\ \left\{\dot{\omega}_z m_k\right\} = \left\{Q_s\right\} \qquad (5\text{-}2)
\end{aligned}
$$

A comparison of Equations (5-1) and (5-2) will aid in determining which terms are included in each of the shorthand matrix expressions.

Each matrix in Equation (5-2) is formulated in a particular section or entry of subroutine FLEX1. The subroutine further manipulates the equation

into a set of linear, simultaneous equations with $\ddot{q}$ as the independent variable.

$$[M] \{\ddot{q}\} = \{B\} \tag{5-3}$$

This set of equations is then solved for $\ddot{q}$.

The flow diagrams that follow indicate the sequential approach used to formulate the components of the flexible body equations of motion and the subsequent solution of these equations. A section of subroutine coding is listed preceded by an explanation of the function of that section. The notation used in the coding is given in Figure (5-1).

A complete listing of the modified TOLA computer program with the flexible body option is given in Appendix A.

m_s GMASS1	GENERALIZED MASS OF s^{th} MODE
ω_s GFREQ	NATURAL FREQUENCY OF s^{th} MODE
Q^k_{xs}, Q^k_{ys}, Q^k_{zs} SXMOD, SYMOD, SZMOD	X, Y, Z COMPONENTS OF THE MODAL DEFLECTIONS FOR THE k^{th} STRUT ATTACH POINT
m_k SMASS	MASS OF k^{th} STRUT
$\ddot{X}$, $\ddot{Y}$, $\ddot{Z}$	RIGID BODY TRANSLATIONAL ACCELERATIONS
a_{k11}, a_{k13}, a_{k31}, a_{k33} A11, A13, A31, A33	DIRECTION COSINES RELATING STRUT COORDINATE SYSTEM TO BODY COORDINATE SYSTEM
$\ddot{S}_k$ SDD	ACCELERATION OF k^{th} STRUT
R_{ksx}, R_{ksy}, R_{ksz}	X, Y, Z COMPONENTS OF STRUT ATTACH POINT POSITION VECTOR
r_{Fk}	DISTANCE FROM STRUT ATTACH POINT TO EXTENDED POSITION OF AXLE
S_{kc}	DISTANCE FROM STRUT AXLE TO STRUT CENTER OF MASS
$\dot{\omega}_x$, $\dot{\omega}_y$, $\dot{\omega}_z$	RIGID BODY ANGULAR ACCELERATIONS
Q_s	GENERALIZED FORCES

FIGURE 5-1 DEFINITION OF SYMBOLS

ENTRY FLEX1

ESTABLISH COMMON REGION AND DIMENSION LOCAL ARRAYS

```
 COMMON/DIRCOM/SKIPUP,DUC(7),AMINER,DM1(13),AA77P,DM2(81),ALA77F,
*DM3(35),AMA77F,DM4(9),ANAZ7F,ANA77P,DM5(12),AX77F,DM6,AY77F,
*DM7(2),AZ77F,DM8(810),YA77P,DM9(188),PI77R,DM10,PI77R1,DM11(5),
*QI77R,DM12,QI77R1,DM13(5),RI77R,DM14,RI77R1,DM15(15),
*U777F,DM16(19),V777F,DM17(3),W777F,DM18(93),NSTRUT,
*MASS(5),RX(5),RY(5),RZ(5),DM19(32),RF(5),DM20(97),SD21(2),
*SD22(2),SD23(2),SD24(2),SD25(2),DM21(99),IN,DM22(153),
*INDLG,DM23(151),INDFLX,NMODE,GMASS1(20),GFREQ(20),
*SXMOD(100),SYMOD(100),SZMOD(100),SKC(5),TXMOD(30),
*DCMODE(60),ARMCDE(120),NPTS,OUTMOD(120),ROIS(60),
*PF(120),GQ(20),GQD1(20),GQD2(20),DDM24(20)
 COMMON/FLXOP/GFORC2(100),GFORC3(100),GFORC4(100),
*T(5),FCX,FCY,FCZ,XD2F(20),XD2T(20),YD2F(20),YD2T(20),ZD2F(20),
*ZD2T(20),XD1F(20),XD1T(20),YD1F(20),YD1T(20),ZD1F(20),ZD1T(20),
*XD0F(20),YD0F(20),ZD0F(20)
 COMMON/LGE/A11(5),A13(5),A31(5),A33(5),DM24(279)
 COMMON/HTCOM/HT,DOMM(2)
 REAL MASS
 DIMENSION GMASS(400),SMASS(25),GSMOD(400),COPMAS(20,20),
*QS(20),QS1(20),GFORC1(60),DIFF(5),RZA11(25),RXA13(25),RKSY(25),
*SDD(10),VARY1(3),OMXD1M(5),OMYD1M(5),OMZD1M(5),GF(20),FDC(3),
*CTMP1(120),GTF(20),SD0(3),SD1(3),SD2(3),TITL(16),LA(40),
*COEF(20,20)
 EQUIVALENCE (SD21(1),SDD(1))
 DATA HDR/6H  FLEX/,(TITL(I),I=1,16)/5HPOINT,4HXD2F,4HXD2T,4HYD2F,
*4HYD2T,4HZD2F,4HZD2T,4HXD1F,4HXD1T,4HYD1F,4HYD1T,4HZD1F,
*4HZD1T,4HXD0F,4HYD0F,4HZD0F/
```

INITIALIZE VARIABLES

```
    DO 1 I=1,400
 1  GMASS(I)=0.
    DO 2 I=1,25
    SMASS(I)=0.
    RZA11(I)=0.
    RXA13(I)=0.
 2  RKSY(I)=0.
    DO 15 I=1,40
 15 LA(I)=0
    NMODE=1
```

RETURN

ORIGINAL PAGE IS
OF POOR QUALITY

ENTRY FLEX2

TEST OPTION INDICATOR

```
IF(INDFLX.EQ.0.OR.INDLG.EQ.0)RETURN
```

FORM LA ARRAY USED BY INTEGRATION ROUTINE

```
NVAR=2*NMODE
CALL INUPD(NVAR,LA)
```

DETERMINE THE COUPLED GENERALIZED MASS MATRIX [M]

```
      DO 30 K=1,NMODE
      NA=(K-1)*NMODE+K
30    GMASS(NA)=GMASS1(K)
      DO 31 N=1,NSTRUT
      LLA=(N-1)*NSTRUT+N
31    SMASS(LLA)=MASS(N)
      CALL GTPRD(SXMOD,SMASS,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GMPRD(CTMP1,SXMOD,GSMOD,NMODE,NSTRUT,NMODE)
      CALL GMADD(GSMOD,GMASS,GMASS,NMODE,NMODE)
      CALL GTPRD(SYMOD,SMASS,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GMPRD(CTMP1,SYMOD,GSMOD,NMODE,NSTRUT,NMODE)
      CALL GMADD(GSMOD,GMASS,GMASS,NMODE,NMODE)
      CALL GTPRD(SZMOD,SMASS,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GMPRD(CTMP1,SZMOD,GSMOD,NMODE,NSTRUT,NMODE)
      CALL GMADD(GSMOD,GMASS,GMASS,NMODE,NMODE)
```

CONVERT MASS MATRIX INTO TRIANGULAR FORM

```
CALL ARAY(1,NMODE,NMODE,20,20,GMASS,COPMAS)
CALL DECOMP(NMODE,COPMAS,COEF)
```

A

A

FORM MATRICES GFORC1, GFORC2, GFORC3, GFORC4

```
      CALL GTPRD(SXMOD,MASS,QS,NSTRUT,NMODE,1)
      CALL GTPRD(SYMOD,MASS,QS1,NSTRUT,NMODE,1)
      CALL CTIE(QS,QS1,CTMP1,NMODE,1,0,0,1)
      CALL GTPRD(SZMOD,MASS,QS,NSTRUT,NMODE,1)
      CALL CTIE(CTMP1,QS,GFORC1,NMODE,2,0,0,1)
      DO 33 L=1,NSTRUT
      DIFF(L)=RF(L)-SKC(L)
      MA=(L-1)*NSTRUT+L
      RZA11(MA)=RZ(L)+DIFF(L)*A11(L)
      RXA13(MA)=RX(L)-DIFF(L)*A13(L)
   33 RKSY(MA)=RY(L)
      CALL GTPRD(SZMOD,RKSY,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GTPRD(SYMOD,RZA11,GSMOD,NSTRUT,NMODE,NSTRUT)
      CALL GMSUB(CTMP1,GSMOD,GFORC2,NMODE,NSTRUT)
      CALL GTPRD(SXMOD,RZA11,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GTPRD(SZMOD,RXA13,GSMOD,NSTRUT,NMODE,NSTRUT)
      CALL GMSUB(CTMP1,GSMOD,GFORC3,NMODE,NSTRUT)
      CALL GTPRD(SYMOD,RXA13,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GTPRD(SXMOD,RKSY,GSMOD,NSTRUT,NMODE,NSTRUT)
      CALL GMSUB(CTMP1,GSMOD,GFORC4,NMODE,NSTRUT)
```

DETERMINE INITIAL DISPLACEMENTS

```
      SUM1=0.
      SUM3=0.
      DO 34 I=1,NSTRUT
      SUM1=SUM1+A31(I)*SDD((I-1)*2+1)
   34 SUM3=SUM3+A33(I)*SDD((I-1)*2+1)
      VARY1(1)=AX77F-SUM1
      VARY1(2)=AY77F
      VARY1(3)=AZ77F-SUM3
      CALL GMPRD(GFORC1,VARY1,QS,NMODE,3,1)
      DO 35 I=1,NSTRUT
      OMXD1M(I)=PI77R1*MASS(I)
      OMYD1M(I)=QI77R1*MASS(I)
   35 OMZD1M(I)=RI77R1*MASS(I)
      CALL GMPRD(GFORC2,OMXD1M,GF,NMODE,NSTRUT,1)
      CALL GMADD(QS,GF,QS,NMODE,1)
      CALL GMPRD(GFORC3,OMYD1M,GF,NMODE,NSTRUT,1)
      CALL GMADD(QS,GF,QS,NMODE,1)
      CALL GMPRD(GFORC4,OMZD1M,GF,NMODE,NSTRUT,1)
      CALL GMADD(QS,GF,QS,NMODE,1)
```

B

B

```
      CALL GTPRD(TXMOD,T,GF,IN,NMODE,1)
      CALL GMSUB(GF,QS,QS,NMODE,1)
      FOC(1)=FCX
      FOC(2)=FCY
      FOC(3)=FCZ
      CALL GTPRD(DCMODE,FOC,GF,3,NMODE,1)
      CALL GMADD(GF,QS,QS,NMODE,1)
      DO 36 I=1,NMODE
      NN=(I-1)*6+1
      CTMP1(NN)=-ARMODE(NN)*AA77P
      CTMP1(NN+1)=ARMODE(NN+1)*YA77P
      CTMP1(NN+2)=-ARMODE(NN+2)*ANA77P
      CTMP1(NN+3)=ARMODE(NN+3)*ALA77F
      CTMP1(NN+4)=ARMODE(NN+4)*AMA77F
36    CTMP1(NN+5)=ARMODE(NN+5)*ANAZ7F
      DO 37 II=1,NMODE
      GF(II)=0.
      DO 37 JJ=1,6
      MJ=(II-1)*6+JJ
37    GF(II)=GF(II)+CTMP1(MJ)*PF(MJ)
      CALL GMADD(GF,QS,QS,NMODE,1)
      DO 38 IG=1,NMODE
      GTF(IG)=GMASS1(IG)*GFREQ(IG)**2.
38    GQ(IG)=QS(IG)/GTF(IG)
```

TRANSFER DISPLACEMENTS TO INTEGRATION ROUTINE

```
      SKIPUP=.TRUE.
      DO 39 I=1,NMODE
      N22=2*I
      CALL UPDAT(1,LA(N22),GQ(I),DU,DU,DU,DU)
39    CONTINUE
      SKIPUP=.FALSE.
      RETURN
```

RETURN

ENTRY FLEX3

TEST OPTION INDICATOR

```
IF(INDFLX.EQ.0.OR.INDLG.EQ.0)RETURN
```

↓

FORM TIME DEPENDENT COMPONENTS OF GFORC1, GFORC2, GFORC3, GFORC4 PRODUCTS

```
    SUM1=0.
    SUM3=0.
    DO 40 I= 1,NSTRUT
    SUM1=SUM1+A31(I)*SDD((I-1)*2+1)
 40 SUM3=SUM3+A33(I)*SDD((I-1)*2+1)
    VARY1(1)=AX77F-SUM1
    VARY1(2)=AY77F
    VARY1(3)=AZ77F-SUM3
    CALL GMPRD(GFORC1,VARY1,GTF,NMODE,3,1)
    DO 46 I=1,NMODE
 46 QS(I)=0.
    CALL GMSUB(QS,GTF,QS,NMODE,1)
    DO 41 I=1,NSTRUT
    OMXD1M(I)=PI77R1*MASS(I)
    OMYD1M(I)=QI77R1*MASS(I)
 41 OMZD1M(I)=RI77R1*MASS(I)

    CALL GMPRD(GFORC2,OMXD1M,GTF,NMODE,NSTRUT,1)
    CALL GMSUB(QS,GTF,QS,NMODE,1)
    CALL GMPRD(GFORC3,OMYD1M,GTF,NMODE,NSTRUT,1)
    CALL GMSUB(QS,GTF,QS,NMODE,1)
    CALL GMPRD(GFORC4,OMZD1M,GTF,NMODE,NSTRUT,1)
    CALL GMSUB(QS,GTF,QS,NMODE,1)
```

↓

CALCULATE GENERALIZED THRUST FORCES

```
 CALL GTPRD(TXMOD,T,GTF,IN,NMODE,1)
 CALL GMADD(QS,GTF,QS,NMODE,1)
```

↓

C

C

CALCULATE GENERALIZED DRAG CHUTE FORCES

```
FDC(1)=FCX
FDC(2)=FCY
FDC(3)=FCZ
CALL GTPRD(DCMODE,FDC,GTF,3,NMODE,1)
CALL GMADD(QS,GTF,QS,NMODE,1)
```

CALCULATE GENERALIZED AERODYNAMIC FORCES

```
    DO 42 I=1,NMODE
    NN=(I-1)*6+1
    CTMP1(NN)=-ARMODE(NN)*AA77P
    CTMP1(NN+1)=ARMODE(NN+1)*YA77P
    CTMP1(NN+2)=-ARMODE(NN+2)*ANA77P
    CTMP1(NN+3)=ARMODE(NN+3)*ALA77F
    CTMP1(NN+4)=ARMODE(NN+4)*AMA77F
42  CTMP1(NN+5)=ARMODE(NN+5)*ANA77F
    DO 43 II=1,NMODE
    GTF(II)=0.
    DO 43 JJ=1,6
    MJ=(II-1)*6+JJ
43  GTF(II)=GTF(II)+CTMP1(MJ)*PF(MJ)
    CALL GMADD(QS,GTF,QS,NMODE,1)
```

FORM THE GENERALIZED STIFFNESS FORCES

```
    DO 44 IG=1,NMODE
    GTF(IG)=Q1(IG)*GMASS1(IG)*GFREQ(IG)**2.
44  QS(IG)=QS(IG)-GTF(IG)
```

SOLVE FOR THE GENERALIZED ACCELERATION

```
CALL SOLVE(NMODE,COEF,QS,GQD2)
```

D

D

TRANSFER GENERALIZED DISPLACEMENTS AND ACCELERATIONS
TO INTEGRATION ROUTINE

```
      DO 45 I=1,NMODE
      NCON=(I-1)*2+1
      MCON=NCON+1
      CALL INTEG(LA(NCON),GQD2(I))
      CALL INTEG(LA(MCON),GQD1(I))
   45 CONTINUE
```

RETURN

ENTRY FLEX4

RETURN

ENTRY FLEX5

RETURN

ENTRY FLEX6

TEST OPTION INDICATOR

```
IF(INDFLX.EQ.0.OR.INDLG.EQ.0)RETURN
```

E

E

TEST IF OUTPUT IS DESIRED

IF (NPTS.EQ.0) RETURN

FORMULATE AND PRINT OUTPUT DATA

```
      CALL STFL(0,1,HDR)
      CALL STFL(2,16,TITL)
      DO 62 I=1,NPTS
      DO 61 K=1,3
      SD0(K)=0.
      SD1(K)=0.
      SD2(K)=0.
      DO 61 J=1,NMODE
      NCON=((I-1)*3+K-1)*NMODE+J
      SD0(K)=SD0(K)+OUTMOD(NCON)*GQ(J)
      SD1(K)=SD1(K)+OUTMOD(NCON)*GQD1(J)
   61 SD2(K)=SD2(K)+OUTMOD(NCON)*GQD2(J)
      LX=(I-1)*3+1
      LY=LX+1
      LZ=LY+1
      XD2F(I)=SD2(1)
      XD2T(I)=AX77F+XD2F(I)+2.*(QI77R*SD1(3)-RI77R*SD1(2))+
     *(PI77R*RI77R+QI77R1)*(ROIS(LZ)+SD0(3))+(PI77R*QI77R
     *-RI77R1)*(ROIS(LY)+SD0(2))-(QI77R*QI77R+RI77R*RI77R)
     * *(ROIS(LX)+SD0(1))
      YD2F(I)=SD2(2)
      YD2T(I)=AY77F+YD2F(I)+2.*(RI77R*SD1(1)-PI77R*SD1(3))+
     *(QI77R*PI77R+RI77R1)*(ROIS(LX)+SD0(1))+(QI77R*RI77R
     *-PI77R1)*(ROIS(LZ)+SD0(3))-(RI77R*RI77R+PI77R*PI77R)
     * *(ROIS(LY)+SD0(2))
      ZD2F(I)=SD2(3)
      ZD2T(I)=AZ77F+ZD2F(I)+2.*(PI77R*SD1(2)-QI77R*SD1(1))+
     *(RI77R*QI77R+PI77R1)*(ROIS(LY)+SD0(2))+(RI77R*PI77R
     *-QI77R1)*(ROIS(LX)+SD0(1))-(PI77R*PI77R+QI77R*QI77R)
     * *(ROIS(LZ)+SD0(3))
      XD1F(I)=SD1(1)
      XD1T(I)=U777F+XD1F(I)+QI77R*(ROIS(LZ)+SD0(3))-RI77R*
     *(ROIS(LY)+SD0(2))
      YD1F(I)=SD1(2)
      YD1T(I)=V777F+YD1F(I)+RI77R*(ROIS(LX)+SD0(1))-PI77R*
     *(ROIS(LZ)+SD0(3))
      ZD1F(I)=SD1(3)
      ZD1T(I)=W777F+ZD1F(I)+PI77R*(ROIS(LY)+SD0(2))-QI77R*
     *(ROIS(LX)+SD0(1))
      XD0F(I)=SD0(1)
      YD0F(I)=SD0(2)
      ZD0F(I)=SD0(3)
      PTN=FLOAT(I)
      CALL STOVAR(8,PTN,XD2F(I),XD2T(I),YD2F(I),YD2T(I),ZD2F(I),
     *ZD2T(I),XD1F(I))
      CALL STOVAR(8,XD1T(I),YD1F(I),YD1T(I),ZD1F(I),ZD1T(I),XD0F(I),
     *YD0F(I),ZD0F(I))
   62 CONTINUE
```

RETURN

ENTRY FLEX7

TEST OPTION INDICATOR

```
IF(INDFLX.EQ.0.OR.INDLG.EQ.0)RETURN
```

TRANSFER VARIABLES TO AND FROM
INTEGRATION ROUTINE

```
      RVAR=2.*FLOAT(NMODE)/4.
      K=IFIX(RVAR)
      IF (K.EQ.0) GO TO 71
      DO 70 I=1,K
      N=4*I-3
      MD1=2*I-1
      MD2=2*I
      CALL UPDAT(4,LA(N),GQD1(MD1),GQ(MD1),GQD1(MD2),GQ(MD2),DU)
 70   CONTINUE
      L=2*K
      IF(L.EQ.NMODE)RETURN
 71   M=4*K+1
      MP=2*K+1
      CALL UPDAT(2,LA(M),GQD1(MP),GQ(MP),DU,DU,DU)
```

RETURN

END

SUBROUTINE DECOMP

TRANSFORM COEFFICIENT MATRIX INTO UPPER AND
LOWER TRIANGULAR FORM
CALLED FROM FLEX2

```
      DIMENSION A(20,20), UL(20,20), SCALES(20), IPS(20)
      COMMON IPS
```

IF (NN.GT.1)

NO

```
      UL(1,1)=A(1,1)
      RETURN
```

YES

SCALE COEFFICIENT MATRIX

```
      DO 5 I=1,N
      IPS(I)=I
      ROWNRM=0.0
      DO 2 J=1,N
      UL(I,J)=A(I,J)
      IF (ROWNRM-ABS(UL(I,J))) 1,2,2
1     ROWNRM=ABS(UL(I,J))
2     CONTINUE
      IF (ROWNRM) 3,4,3
3     SCALES(I)=1.0/ROWNRM
      GO TO 5
4     CALL SING(1)
      SCALES(I)=0.0
5     CONTINUE
```

F

F

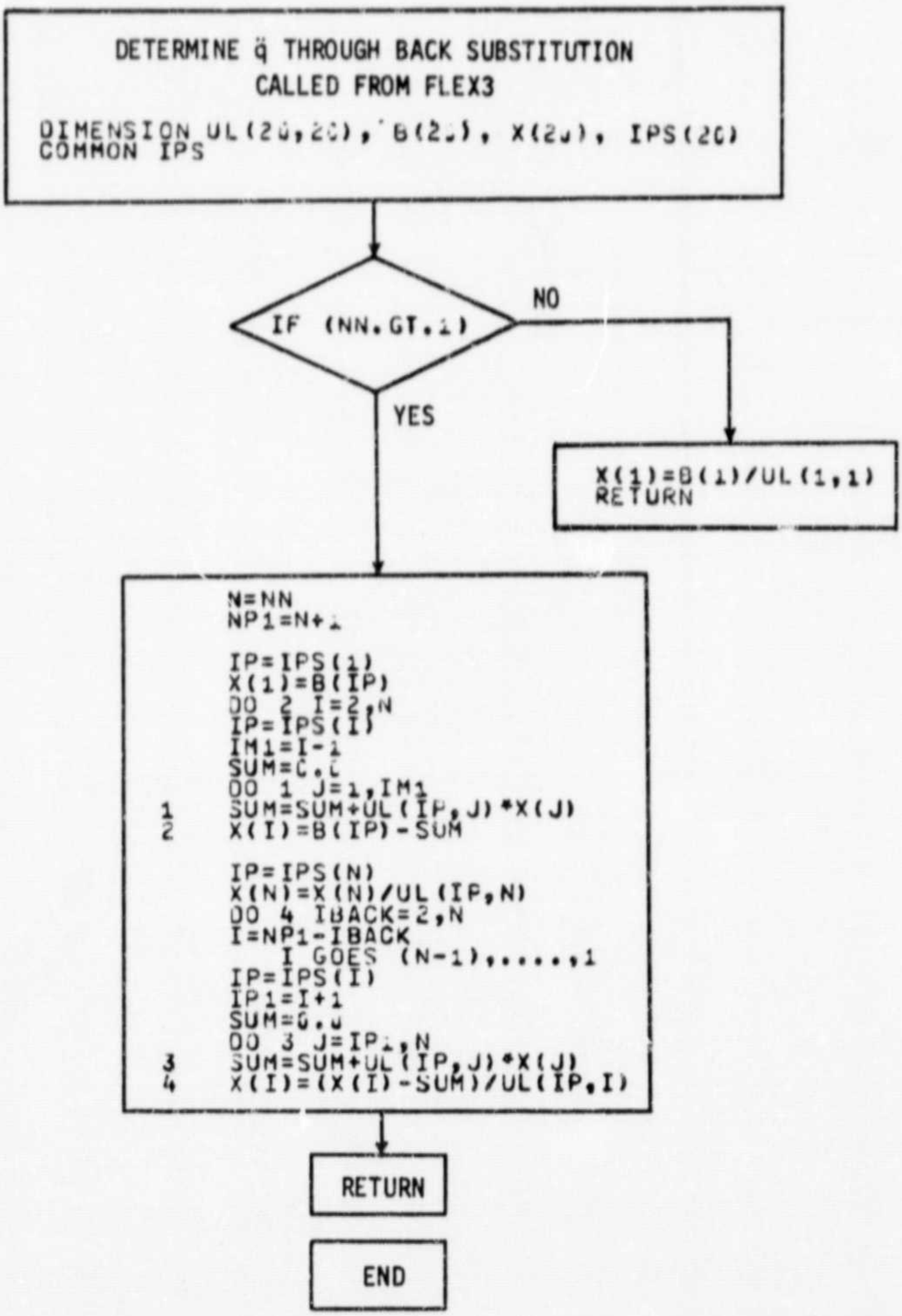
FORMULATE UPPER AND LOWER MATRICES

```
      NM1=N-1
      DO 17 K=1,NM1
      BIG=0.0
      DO 11 I=K,N
      IP=IPS(I)
      SIZE=ABS(UL(IP,K))*SCALES(IP)
      IF (SIZE-BIG) 11,11,10
10    BIG=SIZE
      IDXPIV=I
11    CONTINUE
      IF (BIG) 13,12,13
12    CALL SING(2)
      GO TO 17
13    IF (IDXPIV-K) 14,15,14
14    J=IPS(K)
      IPS(K)=IPS(IDXPIV)
      IPS(IDXPIV)=J
15    KP=IPS(K)
      PIVOT=UL(KP,K)
      KP1=K+1
      DO 16 I=KP1,N
      IP=IPS(I)
      EM=-UL(IP,K)/PIVOT
      UL(IP,K)=-EM
      DO 16 J=KP1,N
      UL(IP,J)=UL(IP,J)+EM*UL(KP,J)
16    CONTINUE
17    CONTINUE
      KP=IPS(N)
      IF (UL(KP,N)) 19,18,19
18    CALL SING(2)
```

RETURN

END

SUBROUTINE SOLVE



SUBROUTINE SING

PRINT DIAGNOSTICS AND TERMINATE PROGRAM IF
COEFFICIENT MATRIX IS ILL-CONDITIONED
CALLED FROM DECOMP

```
11   FORMAT(54H.MATRIX WITH ZERO ROW IN DECOMPOSE.                  )
12   FORMAT(54H.SINGULAR MATRIX IN DECOMPOSE. ZERO DIVIDE IN SOLVE.   )
     NOUT=6
       NOUT=STANDARD OUTPUT UNIT
     GO TO (1,2),IWHY
1    WRITE (NOUT,11)
     GO TO 10
2    WRITE (NOUT,12)
10   CALL EXERR(0)
```

RETURN

END

6. EXAMPLES

This section provides two examples illustrating the modified TOLA computer program input and results. It is intended to show actual program operation with emphasis placed on properly interpreting and incorporating flexible body data. Each example consists of a brief explanation with a listing of all input data and sample output given in the appendix section. Input data for both cases were supplied by the NASA Langley Research Center.

6.1 Rigid Body Example - This case demonstrates how to exercise the modified TOLA computer program without including the effects of flexibility. All the data defined in Reference 1 for a rigid body case are required. As can be seen in the input data listing of Appendix D, setting the indicator INDFLX to zero is the only flexible body data required when running a rigid body case.

6.2 Flexible Body Example - This case demonstrates how to incorporate the effects of flexibility in a takeoff or landing analysis through the use of the flexible body option. All modal data required by the option was supplied by NASA and are shown in Figure 6-1. These data consist of a modal frequency, generalized mass, and modal deflections for sixteen free-free normal modes.

Several observations can be made that help in interpreting program input and results. Only modal deflections in the vertical (Z) direction are given, consequently, there is no flexible body response in the X and Y direction. In addition, the mode shapes defined by the deflections are all symmetric from wing tip to wing tip. This results in a symmetric landing (simultaneous main gear touchdown) since all the rigid body rigid body data are also symmetric.

The first four modes shown in Figure 6-1 were used to represent airframe elasticity. Referring to the input data listing in Appendix E, the flexible body option indicator is set equal to one. Each modal frequency and generalized mass are placed in arrays GFREQ and GMASS1 respectively. The vertical deflections for each landing gear attach point are in SZMOD. Since there are no other landing gear data, SXMOD and SYMOD are not input. Modal data for the aerodynamic reference point are contained in ARMODE. Aerodynamic weighting factors were calculated using the procedure outlined in Appendix B assuming an elliptical spanwise lift distribution over the aircraft's wing. These factors are in array PF. Engine attach point modal data in the X direction are assumed zero; thereforce, TXMODE is not input. Flexible body responses are output for four points

MODE NUMBER	MODAL FREQ HZ	MODAL MASS SLUGS	NORMALIZED MODAL DATA - POSITIVE DOWN						
			PILOT STA.	NOSE GEAR	AERO DEF	AERO SLOPE	MAIN GEARS	ENGINE	TAIL
1	2.099	66.6617	+.1970	+.0800	-.1140	-4.239E-4	-.1000	-.0400	+.2030
2	2.628	79.4557	-.0385	-.0200	-.0414	3.643E-4	-.0042	+.1885	-.2440
3	4.784	70.8341	+.0035	-.1530	+.0405	-6.03E-4	+.1393	+.1380	-.1072
4	6.907	18.1131	+.0035	-.0650	+.0644	-3.034E-5	+.0112	-.0510	-.1250
5	7.671	20.0656	+.0254	+.0400	-.0342	-5.653E-5	-.0087	+.0111	+.1998
6	9.728	99.7316	+.3028	+.1850	-.0054	-1.351E-1	+.1200	+.0559	-.4409
7	11.797	70.4027	-.1673	-.0100	-.1258	3.716E-4	+.1050	+.0588	-.0165
8	13.878	31.6212	-.1157	+.0500	-.0323	-9.602E-4	-.0100	+.0350	-.2167
9	15.552	40.4199	+.0494	-.0300	-.0036	7.309E-4	+.0500	+.1025	+.2985
10	17.638	12.1885	-.0038	+.0050	-.0044	8.596E-7	+.0500	+.0100	+.0982
11	20.019	27.2611	+.0370	-.0500	-.0814	5.78E-4	-.0870	-.0200	-.3728
12	21.099	25.7565	+.0181	-.0350	-.0653	2.801E-4	+.0200	+.0075	-.0837
13	23.396	37.8456	+.0135	-.0500	+.0085	1.617E-4	.0000	+.0400	-.1269
14	23.969	39.6848	-.0180	+.0400	-.0009	-1.503E-4	+.0350	+.0250	-.2163
15	25.637	29.5837	-.0072	+.0150	+.0174	-1.417E-4	-.1300	-.0300	+.2243
16	25.694	8.2589	-.0023	+.0075	+.0044	-4.213E-5	-.0320	+.0050	+.0670

PILOT STATION	X = 42.6667	ENGINES	X = -5.4722
NOSE GEAR	X = 35.08333		Y = ±14.1667
MAIN GEARS	X = -2.83867	TAIL	X = -32.333
	Y = ±8.3333		
	Z = 0.509583		

FIGURE 6-1 FLEXIBLE BODY DATA

on the aircraft, the pilot station, nose gear, right main gear, and tail. Modal data for these points are in OUTMOD and, their position vectors are defined in ROIS.

Sample output for this run follows the input listing.

7. REFERENCES

1. Lynch, Urban H. D., and Dueweke, John J., "Takeoff and Landing Analysis Computer Program (TOLA); Part III - User's Manual," Air Force Flight Dynamics Laboratory Technical Report AFFDL-TR-71-155, April 1974.
2. Young, Fay O., and Dueweke, John J., "Takeoff and Landing Analysis Computer Program (TOLA); Part IV - Programmer's Manual," Air Force Flight Dynamics Laboratory Technical Report AFFDL-TR-71-155, January 1975.
3. Dick, J. W., and Benda, B. J., "Addition of Flexible Body Option to the TOLA Computer Program, Part I - Final Report", NASA CR-132732, October 1975.

APPENDIX A

PROGRAM LISTING - TOLA COMPUTER PROGRAM

WITH FLEXIBLE BODY OPTION

```
      OVERLAY(TOLA,0,0)
      PROGRAM TOLA(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE13,TAPE16,TAPE31)
      COMMON/TABDIR/TABLE(800)
      COMMON/READ1/DM1(64),JBC,INXQ
      CALL FTN3IN(1,0,DUMMY)
      REWIND 13
C     READ SUBROUTINE INITIALIZATION
      JBC=-1
      INXQ=0
 1    CALL EXE
      GO TO 1
      END
      SUBROUTINE EXE
C             EXECUTIVE PROGRAM
      DIMENSION IMTHD(1)
      LOGICAL INDARC , END , SKIPUP
      LOGICAL STSWCH , TABSTP
      COMMON  /STOPIT/ STSWCH , TABSTP
      COMMON /DIRCOM/
     *        SKIPUP       ,H0           ,T            ,TIMINT       ,DM1(4)       ,
     *        AMINER       ,AMAXER       ,
     *RELER1,DM7(5),RELER2,DM2(7),AINCRS(4),DM4(46),TIMER,DM3(64),
     *        AMAXH        ,DM11  (180),DECRES(   4),
     *        DM13   (   8),DELTS        ,DM14   (   6),DLH1         ,DM15   (   4),
     *DLTM1,DLTM2,DLTM3,DM16(18),EPS1,DM8,
     *        EPS11        ,DM21         ,EPS13        ,DM22  ( 12),EPS25        ,
     *        DM23         ,EPS3         ,DM24         ,EPS5         ,DM25         ,
     *        EPS7         ,DM26         ,EPS9         ,DM27  ( 25),GREFF        ,
     *DM28(17),HCC7F,DM29(44),
     *        INDALR       ,DM31   (   2),INDARC       ,DM32         ,
     *INDATM,INDAUT,DM33(123),INDPSK,DM38(34)
      COMMON /DIRCOM/
     *        INDSKP       ,DM4.         ,INDSTE       ,INDSTF       ,INDSTG       ,
     *        DM41   (   2),INDSTR       ,INDSTY       ,DM42         ,INDTFF       ,
     *        DM43         ,INDVPC       ,DM44   (   9),
     *        NSTAGE       ,DM46   (   3),DMGPR        ,DM47  ( 63),REP    ( 30),
     *        RE77F        ,DM48   ( 28),RP77F        ,DM50  ( 14),
     *DM51(6),SIGN,DM52(56),TIME,DM9(2),
     *        TIMES        ,DM55   (   3),TIMSX        ,DM56         ,TL1          ,
     *        TL11         ,TL12         ,TL13         ,TMAX         ,DM57  (127),
     *        PRINT        ,DM58   (   6),STGVAR(   2),DM59  ( 13),
     *DM60(277),NSTRUT,DM63(89),IL,DM62(215),ALPDES,DM61(111),
     *INDLG,DM12(151),DDM(2.68)
      COMMON/TABSRC/DUMM2(11.)
      COMMON/EXEAUT/ADDIM1,SWT1,TIME1,ALPDD1,DUMM1(5)
      COMMON/LG/IDM3(7)
      COMMON/AUTSC/IDM4(40)
      COMMON/AUTPRC/IDM5(53)
      COMMON/LGAUTS/IDM6(14)
      COMMON/FLXUP/IDM1(608)
      COMMON/AUTSAC/ALPDR1,QDESR,DUMM3(4)
      COMMON/HTCOM/HT,HT1,HT2
      COMMON/CONTRO/ICONTR,CONTR1,ICONT2,ICONT3
      LOGICAL SWT1,SWT2,SWT3
      EQUIVALENCE (IMTHD , SKIPUP)
      DATA STCOM2,STCOM3,BLANK/4HTMAX,5HSTAGE,6H      /
 1    FORMAT (1H0,15X,10A6/(1H ,15X,10A6))
 3    FORMAT (1H0,15X,4HSTOP)
 7    FORMAT (1H0,15X,16HSTAGE DATA ERROR)
 9    FORMAT (1H0,15X,11H STAGE ON--,2A6)
 10   FORMAT (1H0,15X,5HSTOP- A6)
C     ************************************************************
C     INITIALIZATION BEFORE DATA READ FOR ALL SUBPROGRAMS.
C     ************************************************************
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 35 II=1,3959
35    IRTHO(II)=0
      DO 36 II=1,30
36    REM(II)=BLANK
      DO 100 II=1,4
      AINCRS(II)=BLANK
100   DECRES(II)=BLANK
      DO 20 I=1,40
20    IDM4(I)=0
      DO 21 I=1,63
21    IDM5(I)=0
      DO 23 I=1,14
23    IDM6(I)=0
      DO 24 I=1,608
 24   IBH1(I)=0
      CALL LGEAR1
      CALL TFFS1
      CALL VPCS1
      CALL SACS1
      CALL OPT1
      CALL FLEX1
      SWT2=.FALSE.
      SWT3=.TRUE.
      NSTAGE=1
      INDSTE=1
      INDATM=1
      SIGN=1.
      AMAXH=1.E6
      DELTS=.1
      PRINT=1.
      DLTM1=.0001
      DLTM2=.0001
      TIMER=30.
      DLH1=1.
      EPS1=1.
      EPS3=1.
      EPS5=1.
      EPS7=1.
      EPS9=1.
      EPS11=1.
      EPS13=1.
      EPS25=1.
      OMGPP=7.29211508E-5
      RE77F=20925428.
      RP77F=20855965.
      RELER1 = .00007
      RELER2 = .000005
      AMINER = .001
      AMAXER=10000.
      TIMINT = AMINER
      DLTM3=.001
      GREFF=32.174
C     AUTS SUBROUTINE INITIALIZATION
      SWT1=.TRUE.
      ALPDD1=0.
C     AUTS - SACS SUBROUTINES INITIALIZATION
      ALPOR1=0.
      QDESR=0.
C     *********************************
C     POST DATA INITIALIZATION AND PRINT
C     *********************************
 264  ICONTR=1
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
 266  CALL STGTSI
 ?1   CALL DEF
```

```
      INDSKP=0
      H0=DELTS
      IF (TIMINT .LT. AMINER) TIMINT = AMINER
      CALL INPUZ
      IF(SWT2)GO TO 302
C     AUTS SUBROUTINE INITIALIZATION
      ADDIM1=ALPUES
      TIME1=TIME
C
      HT=DELTS
      HT1=HT
      HT2=HT
      SWT2=.TRUE.
 302  T=TIME
      TL1=TIME
      TL11=TIME
      TL12=TIME
      TL13=TIME
      TIMES=TIME-TIMSX
      INJARC = HUC7F .GT. AMAXH
      CALL LINES (13)
      WRITE (6,1) (REM(I),I=1,30)
      IF(INOVPG.EQ.0)GO TO 303
      ICONTR=4
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
 303  IF(INOTFF.EQ.0)GO TO 304
      ICONTR=2
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
 304  IF(INOAEZ.EQ.0)GO TO 305
      ICONTR=3
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
 305  ICONTR=5
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
      IF(INOLG.EQ.0)GO TO 306
      ICONTR=6
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
 306  SKIPUP=.TRUE.
      CALL VPCS2
C     POST DATA INITIALIZATION FOR SUBPROGRAMS
      CALL LGEAR2
      CALL OPT2
      CALL FLEX2
 413  CALL OPT7
      CALL SDFLGP
      CALL OPT6
      IF(INOAUT.EQ.0)GO TO 307
      ICONTR=2
      CALL OVERLAY(4LTOLA,2,0,6HRECALL)
 307  SKIPUP=.FALSE.
      END = .FALSE.
      TIMEA=DELTS+T-1.E-6
      TIMEP=PRINT+T-1.E-6
C     ******************
C     EXECUTIVE PROGRAM
C     ******************
 412  TPD = T+DELTS
      IF (TPD .LT. TMAX) GO TO 413
      DELTS = TMAX-T
      H0    = DELTS
      END = .TRUE.
 413  CALL MIMIN(MIM)
      IF(MIM.NE.0)GO TO 415
      CALL SDFLGP
      IF(INOSTC.EQ.0)GO TO 417
      IF(END)GO TO 417
```

```
      IF(INDSTG.NE.0)GO TO 417
      TIMEA=T+DELTS-1.E-6
      IF(T.LT.TIMEP)GO TO 414
      TIMEP=T+PRINT-1.E-6
      CALL OPT6
      IF(INDAUT.EQ.0)GO TO 416
      ICONTR=2
      CALL OVERLAY(4LTOLA,2,0,6HRECALL)
      GO TO 415
414   IF(INDAUT.EQ.0)GO TO 416
      ICONTR=1
      CALL OVERLAY(4LTOLA,2,0,6HRECALL)
      GO TO 416
415   CALL SDFLGP
      IF(T.LT.TMAX)END=.FALSE.
      IF(INDSTG.EQ.0)GO TO 417
      IF(END)GO TO 417
      IF(INDSTG.NE.0)GO TO 417
      CALL OPT5
      TIMEP=T+PRINT-1.E-6
      IF(T.LT.TIMEA)GO TO 416
      TIMEA=T+DELTS-1.E-6
      IF(INDAUT.EQ.0)GO TO 416
      ICONTR=2
      CALL OVERLAY(4LTOLA,2,0,6HRECALL)
      GO TO 415
417   CALL OPT5
      IF(INDAUT.EQ.0)GO TO 416
      ICONTR=2
      CALL OVERLAY(4LTOLA,2,0,6HRECALL)
416   IF (INDSTF .EQ. 0) GO TO 744
      IF (    END      ) GO TO 743
      IF (INDSTG .EQ. 0) GO TO 412
C     *************
C     STAGING IS REQUIRED
C     *************
511   INDSTG = 0
      CALL LINES (2)
      WRITE (6,9) (STGVAR(I),I=1,2)
524   DO 525 JJ=1,4
      AINCRS(JJ)=BLANK
525   DECRES(JJ)=BLANK
      IF(INDSTF.EQ.0) GO TO 537
      INDSTF=0
      CALL LINES (2)
      WRITE (6,10) STCOM3
      GO TO 744
537   INDSKP=1
      CALL DEF
      ICONTR=1
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
C     STAGE DATA IS READ - TEST MAJOR OR MINOR
543   INDSKP=0
572   IF(INDSTY.NE.0) GO TO 601
      CALL STGTSI
      T=TIME
      H0= DELTS
      SKIPUP = .TRUE.
      IF(SWT3)GO TO 22
      IF(INDLG.NE.2)GO TO 22
      NDEFS=5*NSTRUT
      IF(IL.NE.0)NDEFS=3*NSTRUT
      CALL LNUPD(NDEFS)
      SWT3=.TRUE.
      IF(ISIGN(1,INDLG).GT.0)GO TO 410
```

```
      SWITS=.FALSE.
      INDLG=-INDLG
      ICONTR=6
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
      CALL LGEAR2
      CALL LGEAR3
      CALL FLEX2
      GO TO 41.
601   INDSTY=0
      CALL DEF
      IF (INDSTR.EQ.0) GO TO 604
      INDSTR=0
      TIMSX=T
      NSTAGE=NSTAGE+1
604   IF (ISIGN(1,INDVPC).GT.0) GO TO 605
      CALL VPCS1
      INDVPC=-INDVPC
      ICONTR=4
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
605   IF (ISIGN(1,INDTFF).GT.0) GO TO 608
      CALL TFFS.
      INDTFF=-INDTFF
      ICONTR=2
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
608   IF (ISIGN(1,INDAER).GT.0) GO TO 264
      CALL SACS1
      INDAER=-INDAER
      ICONTR=3
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
      GO TO 264
743   CALL LINES (2)
      WRITE (6,1.) STCOM2
C744           GENERAL STOP
744   CALL LINES (2)
      WRITE (6,3)
      STSWCH=.TRUE.
      CALL LGEASP
      RETURN
      END
      SUBROUTINE INUPD (N,L)
      INTEGER L(1)
      COMMON/DIRCOM/SKIPUP(646),INDSTE(1245),DM(2068)
      COMMON /UPDCAL/ NUM , P( 90),Y( 90)
      IF (NUM+N .LE. 90) GO TO 5
      CALL DEF
      WRITE (6,700)
700   FORMAT(55H NUMBER OF INTEGRATION VARIABLES EXCEEDS MAX LIMIT   90)
      STOP
5     DO 20 I= 1,N
      NUM = NUM +1
20    L(I) = NUM
      RETURN
      END
      SUBROUTINE LNUPD(MN)
      COMMON/UPDCAL/NUM,P(90),Y(90)
      NUM=NUM-MN
      RETURN
      END
      SUBROUTINE INPUZ
      COMMON/UPDCAL/NUM,P(90),Y(90)
      NUM    = 0
9     DO 10 I=1,90
      P(I) = 0.
10    Y(I) = 0.
      RETURN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      END
      SUBROUTINE INTEG(K,XD)
      COMMON/UPDCAL/NUM,P(90),Y(90)
      P(K) = XD
      RETURN
      END
      SUBROUTINE UPDAT (JX1,JX2,XJ1,XJ2,XJ3,XJ4,XJ5)
      COMMON/DIRCOM/SKIPUP,DM(645),INDSTE(1245),DM1(2066)
      COMMON/UPDCAL/NUM,P(90),Y(90)
      LOGICAL SKIPUP
      IF (SKIPUP) GO TO (200,201,202,203,204) , JX1
                  GO TO (100,101,102,103,104) , JX1
104   XJ5 = Y(JX2+4)
103   XJ4 = Y(JX2+3)
102   XJ3 = Y(JX2+2)
101   XJ2 = Y(JX2+1)
100   XJ1 = Y(JX2  )
      RETURN
204   Y(JX2+4) = XJ5
203   Y(JX2+3) = XJ4
202   Y(JX2+2) = XJ3
201   Y(JX2+1) = XJ2
200   Y(JX2  ) = XJ1
      RETURN
      END
      SUBROUTINE MIMIN(MIM)
      LOGICAL SW
      COMMON/DIRCOM/F,DX,X,STOPIT,IVARDH,SW,DM1(2),DXMIN,
     *DXMAX,RELER1,DM2(5),RELER2,DM3(629),INDSTE,DM4(930),
     *PRTMIN,DM5(161),INDLG,DM6(151),DM7(2068)
      COMMON/UPDCAL/N,P(90),Y(90)
      COMMON/HTCOM/HT,HT1,HT2
      DIMENSION YMAX(90),Y0(90),P0(90),S(90),YP(90),Y1(90),Z(90),
     CXK(90,3)
      MIM=0
      ACH=0.
      XF=X+DX
      IF (IVARDH .NE. 0) GO TO 300
      X0=X
      DO 20 I=1,N
20    YMAX(I)=ABS(Y(I))
30    H=AMIN1(HT,HT1,HT2)
40    X=X0
      H=AMIN1(ABS(H),ABS(DXMAX))
      DO 50 I=1,N
50    Y0(I)=Y(I)
      CALL OPT7
      CALL OPT4
      IF(SW.OR.(INDSTE.EQ.0))GO TO 360
      DO 60 I=1,N
      P0(I)=P(I)
60    Y(I)=Y0(I)+0.5*H*P0(I)
      X=X0+0.5*H
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      IF(SW.AND.(H.GT.STOPIT)) GO TO 270
      DO 80 I=1,N
      S(I)=2.0*P(I)+P0(I)
80    Y(I)=Y0(I)+0.5*H*P(I)
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      IF(SW.AND.(H.GT.STOPIT)) GO TO 270
      DO 90 I=1,N
```

```
      S(I)=2.0*P(I)+S(I)
 90   Y(I)=Y0(I)+H*P(I)
      X=X0+H
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      IF(SW.AND.(H.GT.STOPIT)) GO TO 270
      DO 100 I=1,N
100   YP(I)=Y0(I)+H*(P(I)+S(I))/6.
110   X=X0+0.25*H
      DO 120 I=1,N
120   Y(I)=Y0(I)+0.25*H*P0(I)
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      IF(SW.AND.(H.GT.STOPIT)) GO TO 270
      DO 130 I=1,N
      S(I)=2.0*P(I)+P0(I)
130   Y(I)=Y0(I)+0.25*H*P(I)
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      IF(SW.AND.(H.GT.STOPIT)) GO TO 270
      DO 140 I=1,N
      S(I)=2.0*P(I)+S(I)
140   Y(I)=Y0(I)+0.5*H*P(I)
      X=X0+0.5*H
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      IF(SW.AND.(H.GT.STOPIT)) GO TO 270
      DO 150 I=1,N
      Y1(I)=Y0(I)+0.5*H*(P(I)+S(I))/6.
150   Y(I)=Y1(I)
      CALL OPT7
      CALL OPT4
      IF(SW.OR.(INDSTE.EQ.0))GO TO 360
      DO 160 I=1,N
      S(I)=P(I)
160   Y(I)=Y1(I)+0.25*H*P(I)
      X=X0+0.75*H
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      IF(SW.AND.(H.GT.STOPIT)) GO TO 270
      DO 170 I=1,N
      S(I)=2.0*P(I)+S(I)
170   Y(I)=Y1(I)+0.25*H*P(I)
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      IF(SW.AND.(H.GT.STOPIT)) GO TO 270
      DO 180 I=1,N
      S(I)=2.0*P(I)+S(I)
180   Y(I)=Y1(I)+0.5*H*P(I)
      X=X0+H
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      IF(SW.AND.(H.GT.STOPIT)) GO TO 270
      R=0.0
      DO 190 I=1,N
      Y(I)=Y1(I)+0.5*H*(P(I)+S(I))/6.
      ERR =(Y(I)-YP(I))/15.0
      Z(I)=AMAX1(YMAX(I),ABS(Y(I)))
```

```
      R = AMAX1( R,ABS(ERR) / AMAX1( Z(I) ,1.0) )
      IF(RELER1 -R)185,190,190
185   IF(ABS(H).GT.ABS(DXMIN)) GO TO 270
190   YMAX(I)=Z(I)
      IF(INDLG.NE.0)CALL LGDET
      IF(SW)GO TO 360
205   X0=X0+H
      IF(ABS(XF-X0).GT.1.E-10)GO TO 211
210   CALL OPT7
      CALL OPT4
      GO TO 360
211   IF(H.GT.PRTMIN)GO TO 212
      ACH=ACH+H
      IF(ACH.GE.PRTMIN)GO TO 213
      GO TO 220
213   CALL OPT7
      CALL OPT4
      HT=H
      CALL LINES(1)
      WRITE(6,700)HT
700   FORMAT(22H INTEG RTN.      HT = ,E15.8)
      MIM=1
      RETURN
212   ACH=0.
220   IF(R-RELER2  )230,230,240
230   H=H+H
240   IF(ABS(H).LE.ABS(XF-X0)) GO TO 40
250   HT=H
      H=XF-X0
      GO TO 40
270   H=0.5*H
      DO 280 I=1,N
280   YP(I)=Y1(I)
      GO TO 110
300   KF=IFIX(DX/DXMIN)
      H=DX/ABS(FLOAT(KF))
      K=0
305   K=K+1
      DO 310 I=1,N
310   Y0(I)=Y(I)
      CALL OPT7
      CALL OPT4
      IF(SW.OR.(INDSTE.EQ.0))GO TO 360
      X=X+0.5*H
      DO 330 J=1,2
      DO 320 I=1,N
      XK(I,J)=H*P(I)
320   Y(I)=Y0(I)+0.5*XK(I,J)
      CALL OPT7
330   CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      X=X+0.5*H
      DO 340 I=1,N
      XK(I,3)=H*P(I)
340   Y(I)=Y0(I)+XK(I,3)
      CALL OPT7
      CALL OPT4
      IF(INDSTE.EQ.0)GO TO 360
      DO 350 I=1,N
350   Y(I)=Y0(I)+(XK(I,1)+2.*(XK(I,2)+XK(I,3))+H*P(I))/6.
      IF(K.LT.KF) GO TO 305
      X=XF
      IF(INDLG.NE.0)CALL LGDET
      CALL OPT7
      CALL OPT4
```

```
0     CALL LINES(1)
      WRITE(6,730)HT
      RETURN
      END
      SUBROUTINE LGDET
      COMMON/UPDCAL/NUM,P(90),Y(90)
      COMMON/LGDE/LA(20),FC2(5),P2(5),PRES(5),C(5),IPPT,LTPT
      COMMON/DIMCOM/DM1(646),INDSTE(675),NSTRUT,DM2(84),
     CA2(5),IL,S2T(5),ES2(5),
     CDM3(5),MASS2(5),DM5(34),ES(5),SB(5),DM4(131),IB(5),DM6(127),
     CINDLG,DM7(151),DM8(2068)
      COMMON/HTCOM/HT,HT1,HT2
      REAL MASS2
      IF(INDLG.EQ.2)RETURN
      DO 10 I=1,NSTRUT
      K=2*I+NSTRUT
      J=LA(K)
      KK=K-1
      JJ=LA(KK)
      L=KK+2*NSTRUT
      LL=LA(L)
      M=K+2*NSTRUT
      MM=LA(M)
      NN=LA(I)
      HT1=HT
      IF(P(J))76,77,78
76    TTIME=Y(J)/ABS(P(J))
79    IF(TTIME.GE.HT)GO TO 77
      HT1=TTIME
      GO TO 77
78    TTIME=(SB(I)-Y(J))/P(J)
      GO TO 79
77    CONTINUE
      IF(Y(J).GT.(-ES(I)))GO TO 50
      WRITE(6,49)I,I,Y(J)
49    FORMAT(58X,4H-ES(,I1,10H) EXCEEDED/
     C58X,2HS(,I1,4H) = E15.7)
50    IF(Y(J).LE.ES(I))GO TO 51
      IF(Y(J).LE.(SB(I)-ES(I)))GO TO 55
      IF(Y(J).LE.(SB(I)+ES(I)))GO TO 52
      WRITE(6,53)I,I,Y(J)
53    FORMAT(58X,4H ES(,I1,10H) EXCEEDED/
     C58X,2HS(,I1,4H) = E15.7)
52    IF(Y(JJ).LE.0.)GO TO 20
      Y(JJ)=0.
      P(J)=0.
20    IF(P(JJ).LT.0.)GO TO 55
      P(JJ)=0.
      GO TO 55
51    IF(Y(JJ).GE.0.)GO TO 21
      Y(JJ)=0.
      P(J)=0.
21    IF(P(JJ).LT.0.)P(JJ)=0.
55    CONTINUE
      IF(IL.NE.J)GO TO 60
      HT2=HT
      IF(P(MM))86,87,88
86    TTIME=Y(MM)/ABS(P(MM))
89    IF(TTIME.GE.HT)GO TO 87
      HT2=TTIME
      GO TO 87
88    TTIME=(S2T(I)-Y(MM))/P(MM)
      GO TO 89
87    CONTINUE
      IF(Y(MM).GT.(-ES2(I)))GO TO 58
```

```
      WRITE(6,57)I,I,Y(MM)
57    FORMAT(58X,5H-ES2(,I1,10H) EXCEEDED/
     C58X,3HS2(,I1,4H) = E15.7)
58    IF(Y(MM).LE.ES2(I))GO TO 61
      IF(Y(MM).LE.(S2T(I)-ES2(I)))GO TO 60
      IF(Y(MM).LE.(S2T(I)+ES2(I)))GO TO 62
      WRITE(6,63)I,I,Y(MM)
63    FORMAT(58X,5H ES2(,I1,10H) EXCEEDED/
     C58X,3HS2(,I1,4H) = E15.7)
62    IF(Y(LL).LE.0.)GO TO 22
      Y(LL)=0.
      P(MM)=0.
22    IF(P(LL).LT.0.)GO TO 60
      GO TO 14
61    IF(Y(LL).GE.0.)GO TO 23
      Y(LL)=0.
      P(MM)=0.
23    IF(P(LL).GE.0.)GO TO 60
140   P(LL)=0.
60    CONTINUE
      IF(IB(I).NE.(-1))GO TO 10
      P(NN)=0.
      Y(NN)=0.
10    CONTINUE
      RETURN
      END
      SUBROUTINE STGTS1
      LOGICAL SW
      COMMON /DIRCOM/
     *COMVAL(5 ) , SW(19) , AINCRS(4) , DM2(292) , DECRES(4) , DM3(910),
     *STEST ( 4) , STESTD(4) , DM4(187) , STGVAR(4) , DM5(543),
     *ISTAGE,DM6(314),DM7(2068)
      COMMON/STGT/ICOUNT,KCOUNT,LOCAIN(4),LOCADE(4)
      COMMON/CONTRO/ICONTR,CONTR1,ICONT2,ICONT3
      DATA BLANK/1H /
      ICOUNT = 0
      KCOUNT = 0
      DO 10 I=1,4
      IF (AINCRS(I) .EQ. BLANK) GO TO 15
      ICONTR=8
      CONTR1=AINCRS(I)
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
      LOCAIN(I)=ICONT2
      IF(ICONT3.EQ.0)GO TO 12
      WRITE (6,11) AINCRS(I)
11    FORMAT (28H ERROR.THE STAGE VARIABLE **A6,
     *       25H** IS NOT IN THE DIRECTORY/24H0...LOOKING FOR NEW CASE)
      CALL EXERR(0)
      RETURN
12    ICOUNT = ICOUNT+1
10    CONTINUE
15    DO 20 I=1,4
      IF (DECRES(I) .EQ. BLANK) GO TO 21
      ICONTR=8
      CONTR1=DECRES(I)
      CALL OVERLAY(4LTOLA,1,0,6HRECALL)
      LOCADE(I)=ICONT2
      IF(ICONT3.EQ.0)GO TO 24
      WRITE (6,11) DECRES(I)
      CALL EXERR(0)
      RETURN
24    KCOUNT = KCOUNT+1
20    CONTINUE
21    RETURN
      END
```

```
      SUBROUTINE STGTST(INDSTG)
      COMMON/DIRCOM/
     1COMVAL(5),SW,DM1(18),AINCRS(4),DM2(292),DECRES(4),DM3(510),
     2STEST(4),STESTD(4),DM4(187),STGVAR(4),DM5(543),ISTAGE,DM6(314),
     3DM7([illegible])
      COMMON/STGT/ICOUNT,KCOUNT,LOCAIN(4),LOCADE(4)
      LOGICAL SW
      DATA AINCR,ADECR/5HINCR.,5HDECR./
22    IF (ICOUNT.EQ.0) GO TO 30
      IF(ISTAGE.EQ.0)GO TO 61
      DO 60 I=1,ICOUNT
      J=LOCAIN(I)
      IF(COMVAL(J).NE.STEST(I))GO TO 30
60    CONTINUE
      GO TO 40
61    DO 23 I=1,ICOUNT
      J = LOCAIN(I)
      IF (COMVAL(J) .GE. STEST (I) ) GO TO 40
23    CONTINUE
30    IF (KCOUNT .EQ. 0) GO TO 33
      IF(ISTAGE.EQ.0)GO TO 64
      DO 65 I=1,KCOUNT
      J=LOCADE(I)
      IF(COMVAL(J).NE.STESTD(I))GO TO 33
65    CONTINUE
      GO TO 50
64    DO 32 I=1,KCOUNT
      J = LOCADE(I)
      IF (COMVAL(J) .LE. STESTD(I) ) GO TO 50
32    CONTINUE
33    INDSTG = 0
      SW = .FALSE.
      RETURN
40    STGVAR(2)=AINCRS(I)
      STGVAR(1)=AINCR
      GO TO 51
50    STGVAR(2)=DECRES(I)
      STGVAR(1)=ADECR
51    INDSTG = 1
      SW = .TRUE.
      RETURN
      END
      BLOCK DATA TDATA
      COMMON/TABCOM/LOCS(115),STABLE(115)
      DATA (STABLE(I),I=1,115)/
     *6HBTAB01,6HWTAB01,6HWTAB02,6HWTAB03,6HTTAB01,6HTTAB02,6HTTAB10,
     *6HTTAB11,6HCTAB01,6HCTAB02,6HCTAB03,6HCTAB04,6HCTAB05,6HCTAB06,
     *6HFTAB01,6HFTAB02,6HFTAB03,6HVTAB01,
     *6HVTAB02,6HVTAB03,6HVTAB04,6HVTAB05,6HVTAB06,6HVTAB07,6HVTAB08,
     *6HVTAB09,6HVTAB10,6HVTAB11,6HVTAB12,6HVTAB13,6HVTAB14,6HVTAB15,
     *6HVTAB16,            6HVTAB17,6HVTAB18,6HATAB01,6HATAB02,6HATAB03,
     *6HATAB04,6HATAB05,6HATAB06,6HATAB07,6HATAB08,6HATAB09,6HATAB10,
     *6HATAB11,6HATAB12,6HATAB13,6HATAB14,6HATAB15,6HATAB16,6HATAB17,
     *6HATAB18,6HATAB19,6HATAB20,6HATAB21,6HATAB22,6HATAB23,6HATAB24,
     *6HATAB25,6HATAB26,6HATAB27,6HATAB28,6HATAB29,6HATAB30,6HATAB31,
     *6HATAB32,6HATAB33,6HATAB34,6HATAB35,6HATAB36,6HATAB37,6HATAB33,
     *6HATAB39,6HATAB40,6HATAB41,6HATAB42,6HATAB43,6HATAB44,6HATAB45,
     *6HATAB46,6HATAB47,6HATAB48,6HATAB49,6HATAB50,6HATAB51,6HATAB52,
     *6HATAB53,6HATAB54,6HATAB55,6HATAB56,6HATAB57,6HATAB58,6HATAB59,
     *6HATAB60,6HATAB61,6HATAB62,6HATAB63,6HATAB64,6HATAB65,6HATAB66,
     *6HATAB67,6HATAB68,6HATAB69,6HATAB70,6HATAB71,6HATAB72,6HATAB73,
     *6HATAB74,6HATAB75,6HATAB76,6HATAB77,6HATAB78,6HATAB80,6H      /
      END
      SUBROUTINE TSRCH(SYM2,LOC2,N2,IER)
      COMMON/TABCOM/LOCS(115),STABLE(115)


      INTEGER LOC2(N2)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DATA MAXT/115/
      DO 1000 I=1,MAXT
      IF (SYM2 .NE. STABLE(I) ) GO TO 1000
      JM=I
      DO 500 J=1,N2
      LOC2(J) = LOCS(JM)
500   JM=JM+1
      IER = -1
      RETURN
1000  CONTINUE
      IER = 1
      RETURN
      END
      SUBROUTINE ASPCH(LOC1,SYM1)
      COMMON/TABCOM/LOCS(115),STABLE(115)
      DATA BLANK,MAXT/1H ,115/
      DO 2000 I=1,MAXT
      IF (LOC1 .NE. LOCS(I)) GO TO 2000
      SYM1 = STABLE(I)
      RETURN
2000  CONTINUE
      SYM1 = BLANK
      RETURN
      END
      SUBROUTINE DEF
C              TITLE PRINT ROUTINE
      COMMON /DIRCOM/ D1(543) , INDSOF , D2(19) , LONG , NCASE , NPAGE ,
     *NSTAGE,D3(1224),D4(2068)
 900  FORMAT(1H1,38X,40HSIX DEGREES OF FREEDOM FLIGHT PATH STUDY/
     *1H ,42X,28HGENERALIZED COMPUTER PROGRAM/
     *1H0,28X,7HINDSOF I2,5X,5HCASE A6,7X,
     36HSTAGE I2,5X,5HPAGE I6/1H )
      NPAGE=NPAGE+1
      LONG=0
      WRITE(6,900)INDSOF,NCASE,NSTAGE,NPAGE
      RETURN
      END
      SUBROUTINE STFL(JOPT,N,ARG1)
C       GENERAL PRINT ROUTINE
      LOGICAL CLEAN,INTEG
      COMMON/CLEAUP/I2,CLEAN,INTEG
      COMMON/STORA/ARG(48),ALIST(8),GETARG(8),NENT,LENT,K
      DIMENSION ARG1(1)
      IF(JOPT.EQ.0)GO TO 5
      IF(JOPT.EQ.3)GO TO 6
      DO 4 I=1,N
4     ARG(I)=ARG1(I)
5     IF(JOPT-1)1,2,3
1     NENT=1
      IF(I2.EQ.0)RETURN
      CLEAN=.TRUE.
      CALL ARRAY(N,1)
      RETURN
C
2     NENT=2
      K=1
      CALL ARRAY(N,0)
      RETURN
C
3     NENT=4
      K=1
      CALL ARRAY(N,0)
      RETURN
6     GETARG(1)=ARG1(1)
      NENT=5
```

```
      K=2
      CALL ARRAY(N,0)
      RETURN
      END
      BLOCK DATA STFLD
      COMMON/CLEAUP/I2,CLEAN,INTEG
      LOGICAL CLEAN,INTEG
      DATA I2,CLEAN,INTEG/0,.TRUE.,.FALSE./
      END
      SUBROUTINE STOVAR(N,A,B,C,D,E,F,G,H)
      COMMON/STORA/ARG(48),ALIST(8),GETARG(8),NENT,LENT,K
      IF(IABS(N).LE.2)GO TO 2
      IF(IABS(N).LE.4)GO TO 1
      GETARG(8)=H
      GETARG(7)=G
      GETARG(6)=F
      GETARG(5)=E
1     GETARG(4)=D
      GETARG(3)=C
2     GETARG(2)=B
      GETARG(1)=A
      NENT=2
      K=2
      CALL ARRAY(N,0)
      RETURN
      END
      SUBROUTINE ARRAY(N,IOPT)
      LOGICAL CLEAN,INTEG
      COMMON/CLEAUP/I2,CLEAN,INTEG
      COMMON/STORA/ARG(48),ALIST(8),GETARG(8),NENT,LENT,K
705   FORMAT(5X,I8,7I15)
710   FORMAT(7X,A6,7(9X,A6))
715   FORMAT(1X,8(1PE15.7))
720   FORMAT(1H )
      IF(IOPT.NE.0)GO TO 2000
      IF(NENT.EQ.LENT)GO TO 1010
      IF(I2.EQ.0)GO TO 1005
      CLEAN=.TRUE.
      GO TO 2000
1005  CALL LINES (1)
      WRITE (6,720)
      IF (NENT.EQ.1) RETURN
1010  INTEG = (N.LT.0)
      NMAX=IABS(N)
      LENT=NENT
      J=1
      GO TO 500
C
C     CLEAN OUT ARRAY
2000  CALL LINES (1)
      IF (LENT.GE.4) GO TO 2010
      IF (INTEG)       GO TO 2005
      WRITE (6,715) (ALIST(I),I=1,I2)
      GO TO 2020
2005  WRITE (6,705) (ALIST(I),I=1,I2)
      GO TO 2020
2010  WRITE (6,710) (ALIST(I),I=1,I2)
2020  I2=0
      IF ( .NOT.CLEAN) GO TO 500
      CLEAN=.FALSE.
      GO TO 1005
C
C     DEVELOP ARRAY
C
501   ALIST (I2) = GETARG (J)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      GO TO 520
510   ALIST (I2) = ARG    (J)
520   J=J+1
      IF (I2 .EQ. K) GO TO 2000
500   IF (J.GT.NMAX) RETURN
      I2=I2+1
      GO TO (510,501),K
      END
      SUBROUTINE LINES(LCOUNT)
C             LINES ACCOUNTING ROUTINE
      COMMON /DIRCOM/D1(663),LONG,D2(1227),D3(2068)
      LONG=LONG+LCOUNT
      IF(LONG.LE.51)RETURN
      CALL DEF
      LONG=LCOUNT
      RETURN
      END
      FUNCTION ASIN(X)
      EXTERNAL ACOS
      ASIN=1.5707963-ACOS(X)
      RETURN
      END
      FUNCTION ACOS(X)
      IF (ABS(X) .GT. 1.0) GO TO 100
      IF (ABS(X).GT. 7.4505806E-09) GO TO 50
      ACOS=1.5707963
      RETURN
50    IF (X .NE. 1.) GO TO 40
      ACOS=0.
      RETURN
40    IF (X .NE. (-1.)) GO TO 30
      ACOS=3.1415926
      RETURN
30    A=0.
      X1=X
      DO 20 I=1,27
      IF (X1 .GT. 0.)GO TO 10
      SA=1.
      X2=1.-(2.*X1**2)
      GO TO 5
10    SA=0.
      X2=(2.*X1**2)-1.
5     A=A+SA*2.**(-(I-1)-1)
20    X1=X2
      ACOS=3.1415926*A
      RETURN
 100  WRITE (6,200)X
 200  FORMAT(26H0ARGUMENT OUT OF RANGE X =E20.8)
      ACOS=99999999.
      RETURN
      END
      FUNCTION ATAN2 (Y,X)
C     ******************************************************
C     SINGLE PRECISION FORTRAN ARCTANGENT FUNCTION SUBROUTINE
C     NOTE. ------- ZERO ARGUMENTS WILL EXIT WITH ZERO ANSWER
C     ******************************************************
      LOGICAL INDIC
      REAL L
      DIMENSION AA(6),A(4),B(4),PN(4)
      DATA  (AA(I), I = 1,6)  /
C     MIN IS AA(1), MAX IS AA(6)
     *   .180264515E-8, .176326981E+0, .577350269E+0, .119175359E+1,
     2   .274747742E+1, .134217728E+9  /
      DATA  (A(I), I = 1,4)  /
     *   .449587214E+0, .195014224E+0, .944754986E-1, .288535359E-1  /
```

```
      DATA  (B(I), I = 1,4) /
     *   .139385770E+1, .396745266E+0, .218181818E+0, .168724015E+0  /
      DATA (PN(I), I = 1,4)  /
     *   .349065850E+0, .698131701E+0, .104719755E+1, .139626340E+1  /
      DATA C1, C2, C3, L  /
     *   .511134291E-1, .270998425E-2, .216649136E+0, .163636364E+0  /
      DATA  PIOV2, ONEPI  /
     *   .157079633E+1, .314159265E+1  /
      IF (Y .EQ. 0.0)  GO TO 16
      IF (X .EQ. 0.0)  GO TO 15
      INDIC = (X .LT. 0.0)
C     INDIC = .TRUE., ANGLE IS IN QUADRANT 1 OR 4
      R = Y/X
    9 Z = ABS(R)
      DO 10  I = 1,6
      IF (Z-AA(I) .LT. 0.0)   GO TO (18,12,11,11,11,11),I
   10 CONTINUE
      GO TO 17
   11 I = I - 2
      J = 6 - I
      P = PN(I)
      Z = A(I) - B(I)/(Z + AA(J))
      GO TO 13
   12 Z = Z*L
      J = 5 - I
      P = 0.0
   13 T = Z**2 + C3 - C2/((Z**2) + C1)
      ATAN2 = Z/T + P
   14 ATAN2 = SIGN(ATAN2,R)
   19 IF ( .NOT. INDIC)  RETURN
      IF (ATAN2 .GT. 0.0)   GO TO 23
      ATAN2 = ATAN2 + ONEPI
      RETURN
   15 ATAN2 = SIGN(PIOV2,Y)
      RETURN
   16 IF (X .GE. 0.0)  GO TO 22
      ATAN2 = ONEPI
      RETURN
   17 ATAN2 = PIOV2
      GO TO 14
   18 ATAN2 = R
      GO TO 19
   22 ATAN2 = 0.0
      RETURN
   23 ATAN2 = ATAN2 - ONEPI
      RETURN
      END
      SUBROUTINE ERROR(LOCT)
C               GENERAL TABLE ERROR ROUTINE
      COMMON /DIRCOM/ DM(5) , SW(641) , INDSTE , DM1(1244),DM2(2068)
      DATA IBL / 1H  /
      INDSTE=0
      CALL LINES (2)
      CALL ASRCH(LOCT,NAME)
      IF (NAME .EQ. IBL) GO TO 5
      WRITE (6,1) NAME
      RETURN
5     WRITE (6,2)
1     FORMAT (14H0TABLE ERROR  ,A6)
2     FORMAT (12H0TABLE ERROR/1H ,47H...LOCATION OF TABLE NOT LISTED IN
     1DIRECTORY...)
      RETURN
      END
      SUBROUTINE EXERR(NUM)
      COMMON/DIRCOM/DM(5),SW(641),INDSTE,DM1(1244),DM2(2068)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      INDSTE = 0
      IF (NUM .EQ. 0) RETURN
      CALL STFL(0,1,DU)
      CALL LINES (3)
      WRITE (6,3) NUM
3     FORMAT (1HA,15X,12HSTOF NUMBER I3)
      RETURN
      END
      SUBROUTINE NDTLU (ND,NA,X,Z,XA,ZR,IE)
C             N-DIMENSION TABLE LOOK-UP ROUTINE
C                                  ND     DIM. OF LOOK-UP
C                                  NA     NUMBERS OF VALUES OF EACH TABLE
C                                  X      TABLES OF EACH X IN ORDER
C                                  Z     FUNCTION VALUES
C                                  XA     ARRAY OF ARGUMENTS
C                                  ZR     RESULT
C                                  IE     ERROR RETURN
C                                    0    NO ERROR
C                                   -1    X ARRAY TO SMALL
C                                    1    X ARRAY TO LARGE
C                                    2    ARRAY NOT ASCENDING ORDER
C  WJ(2**MAX(N))
      DIMENSION X(1),Z(1),NA(1),XA(1),NS(8),WJ( 32)
      IE=0
      L1=2
      LF=ND-1
      DO 3 I=1,LF
      L2=L1+NA(I)-2
      FIND=0.
      DO 4 J=L1,L2
      IF(X(J).GT.X(J-1)) GO TO 6
      IE=2
      RETURN
6     IF(FIND.NE.0.) GO TO 4
      IF  (XA(I)-X(J-1)) 8,4,4
8     IF  (J.GT.L1) GO TO 10
      IE=-1
      RETURN
10    FIND=1.
      NS(I)=J-2
4     CONTINUE
      IF  (FIND) 11,12,11
12    IF  (XA(I)-X(L2)) 13,13,14
14    IE=1
      RETURN
13    NS(I)=L2-1
11    L1=L2+2
3     CONTINUE
      KF=2**LF
      MW=-2
      DO 18 I=1,KF,2
      L1=0
      IZ=0
      MW=MW+2
      NPT=1
      DO 19 J=1,LF
      MM=2**(J-1)
      IF(AND(MW,MM).EQ.0)GO TO 22
      N=NS(J)+1
      GO TO 23
22    N=NS(J)
23    N=N-L1
      L1=L1+NA(J)
      IZ=NPT*(N-1)+IZ
      NPT=NPT*NA(J)
```

```
      WJ(I)=Z(I2+1)
48    WJ(I+1)=Z(I2+2)
      L1=1
      DO 32 I=1,LF
      M=NS(I)
      PER=(XA(I)-X(M))/(X(M+1)-X(M))
      JF=KF/2
      DO 32 J=1,JF
32    WJ(J)=WJ(2*J-1)+(WJ(2*J)-WJ(2*J-1))*PER
      ZR=WJ(1)
      RETURN
      END
      SUBROUTINE ATMS (HGC7F)
C              ATMOSPHERE CALCULATION ROUTINE
      LOGICAL HGT90,EVEN
      DIMENSION HB(11),RHOB(11),PB(11),TB(11),AK1(11),AK2(11),AK3(11),
     1A(2),B(2),C(2),D(2)
      COMMON /DIRCOM/
     1DM(159)    ,ANUA7F     ,DM1(324)  ,INDATM     ,DM2(132)  ,PA77P     ;
     2DM3(95)    ,RHOAS      ,DM4(89)   ,TA77R      ,DM5(80)   ,VS77F     ;
     3DM6(946),DM7(2068)
C
C         ALTITUDE,    DENSITY , PRESSURE  ,TEMPERATURE,          /
      DATA (HB(I),I=1,11)/0.,1.1E4,2.5E4,4.7E4,5.3E4,7.9E4,
     C9.E4,10.5E4,16.E4,17.E4,2.E5/
      DATA (RHOB(I),I=1,11)/23.7692E-4,7.062E-4,7.765E-5,
     C2.8804E-5,13.9468E-7,4.1189E-8,4.2610E-9,2.232E-10,
     C1.845E-12,1.338E-12,6.113E-13/
      DATA (PB(I),I=1,11)/2116.21695,472.73,51.979,2.5155,
     C1.2181,2.1.8E-2,21.8.9E-4,15.562E-5,75.578E-7,
     C58.954E-7,29.759E-7/
      DATA (TB(I),I=1,11)/518.688,389.988,389.988,508.788,
     C508.188,298.188,298.188,406.188,2386.188,
     C2566.188,2836.188/
      DATA (AK1(I),I=1,11)/-.225569E-4,0.,.138466E-4,0.,
     C-.1592.2E-4,0.,.241458E-4,.886289E-4,.754341E-5,
     C.353715E-5,.222124E-3/
      DATA(AK2(I),I=1,11)/-5.25612,0.,11.3883,0.,-7.59218,
     C0.,8.5412,1.7.824,3.41648,6.83296,9.76137/
      DATA(AK3(I),I=1,11)/0.,.157689E-3,0.,.120869E-3,
     C0.,.200234E-3,5*0./
      DATA A(1),A(2),B(1),B(2),C(1),C(2),D(1),D(2)/.759511,.935787,
     C.174164,.273966,220.,180.,25.,140./
C
      IF ((INDATM.EQ.0).OR.(HGC7F.GE.2.5E6)) GO TO 470
      IF (HGC7F.LE.0.) GO TO 460
      TMP=.3048*HGC7F
      HGP=TMP/(1.+TMP/6356766.)
      HGT90=.TRUE.
      M=2
      IF (HGP.GT.180000.) GO TO 49
      M=1
      IF (HGP.GT.90000.) GO TO 49
      HGT90=.FALSE.
C  SEARCH FOR   LAYER
49    IF (HGP.LT.HB(11)) GO TO 50
      LAY=11
      GO TO 60
50    DO 55 I=2,11
      IF (HGP.GT.HB(I)) GO TO 55
      LAY=I-1
      GO TO 60
55    CONTINUE
60    TMP=HGP-HB(LAY)
C     TMP2=1.+AK1(LAY)*TMP
```

```
      IF (LAY.GT.6) GO TO 69
      IF (2*(LAY/2).NE.LAY) GO TO 69
      EVEN=.TRUE.
      GO TO 70
69    EVEN=.FALSE.
70    TA77R= TB(LAY)*TMP2
      IF (EVEN) GO TO 72
      PA77P=PB(LAY)*TMP2**(-AK2(LAY))
      RHOAS=RHOB(LAY)*TMP2**(-1.-AK2(LAY))
      GO TO 80
72    TMP3=EXP(-AK3(LAY)*TMP)
      PA77P=PB(LAY)*TMP3
      RHOAS=RHOB(LAY)*TMP3
80    IF (HGT9.) GO TO 85
      STA77R=SQRT(TA77R)
      VS77F= 49.02576*STA77R
      ANUA7F=.0226988E-6*(TA77R*STA77R/((TA77R+198.72)*RHOAS))
      RETURN
85    VS77F=0.
      ANUA7F=0.
      TA77R=TA77R*(A(M)-B(M)*ATAN(HGP-C(M)/D(M)))
      RETURN
460   VS77F=1116.43372
      TA77R=TB(1)
      PA77P=PB(1)
      ANUA7F=1.5723288E-4
      RHOAS=RHOB(1)
      RETURN
470   VS77F=0.
      TA77R=0.
      PA77P=0.
      ANUA7F=0.
      RHOAS=0.
      RETURN
      END
      SUBROUTINE INVR3(A,B,INDER)
C               A 3 X 3 MATRIX INVERSION ROUTINE
      DIMENSION A(3,3),B(3,3)
      B(1,1)=A(2,2)*A(3,3)-A(2,3)*A(3,2)
      D=B(1,1)*A(1,1)
      B(1,2)=A(3,2)*A(1,3)-A(1,2)*A(3,3)
      D=D+B(1,2)*A(2,1)
      B(1,3)=A(1,2)*A(2,3)-A(2,2)*A(1,3)
      D=D+B(1,3)*A(3,1)
      IF(ABS(D).LE.1.E-18)GO TO 19
      B(2,1)=A(3,1)*A(2,3)-A(3,3)*A(2,1)
      B(2,2)=A(1,1)*A(3,3)-A(3,1)*A(1,3)
      B(2,3)=A(2,1)*A(1,3)-A(1,1)*A(2,3)
      B(3,1)=A(2,1)*A(3,2)-A(3,1)*A(2,2)
      B(3,2)=A(3,1)*A(1,2)-A(1,1)*A(3,2)
      B(3,3)=A(1,1)*A(2,2)-A(1,2)*A(2,1)
      DO 17 J=1,3
      DO 17 I=1,3
17    B(I,J)=B(I,J)/D
      INDER=1
      RETURN
19    INDER=2
      RETURN
      END
      SUBROUTINE MULT31 (A,B,C)
C               A MATRIX MULTIPLICATION ROUTINE
      DIMENSION A(3,3),B(3),C(3)
      DO 1 I=1,3
      C(I)=0.
      DO 1 J=1,3
```

```
      C(I)=C(I)+A(I,J)*B(J)
      RETURN
      END
      SUBROUTINE MULT33(A,B,C)
C               A MATRIX MULTIPLICATION ROUTINE
      DIMENSION A(3,3),B(3,3),C(3,3)
      DO 1 I=1,3
      DO 1 J=1,3
      C(I,J)=0.
      DO 1 K=1,3
 1    C(I,J)=C(I,J)+A(I,K)*B(K,J)
      RETURN
      END
      SUBROUTINE TRNPOS (A,B)
C               A 3 X 3 MATRIX TRANSPOSE ROUTINE
      DIMENSION A(3,3),B(3,3)
      DO 1I=1,3
      DO 1J=1,3
 1    B(I,J)=A(J,I)
      RETURN
      END
      SUBROUTINE MIMC(N,LOCT,NX1,NX2,NX3,NX4,X1ARG,X2ARG,
     *X3ARG,X4ARG,A)
C               TABLE SETUP ROUTINE
      COMMON/TABDIR/TABLE(1)
      DIMENSION XA(4),NA(4)
 11   FORMAT (1H0,15X,40HAN ARGUMENT EXCEEDS LOWER LIMIT OF TABLE)
 12   FORMAT (1H0,15X,40HAN ARGUMENT EXCEEDS UPPER LIMIT OF TABLE)
 16   FORMAT (1H0,15X,44HINDEPENDENT VARIABLES NOT IN ASCENDING ORDER)
      XA(1)=X1ARG
      XA(2)=X2ARG
      XA(3)=X3ARG
      XA(4)=X4ARG
      NA(1)=NX1
      NA(2)=NX2
      NA(3)=NX3
      NA(4)=NX4
      NAT=NX1
      IF (N-4) 3,4,5
 3    NAT=NAT+NA(2)
      GO TO 6
 4    NAT=NAT+NA(2)+NA(3)
      GO TO 6
 5    NAT=NAT+NA(2)+NA(3)+NA(4)
 6    II=NAT+LOCT
      CALL NDTLU(N,NA,TABLE(LOCT),TABLE(II),XA,A,IERROR)
      IF (IERROR.EQ.0) RETURN
      CALL STFL(0,1,DUM)
      CALL ERROR(LOCT)
      CALL LINES (2)
      IF (IERROR-1) 7,14,15
 7    WRITE (6,11)
 13   RETURN
 14   WRITE (6,12)
      RETURN
 15   WRITE (6,16)
      RETURN
      END
      SUBROUTINE TLU(X,LOCT,Y)
C               TWO DIMENSIONAL TABLE LOOK-UP ROUTINE
      COMMON/TABDIR/C(1)
      EQUIVALENCE (N,Z)
      LOCTM1=LOCT-1
      N=
      Z=C(LOCT)
```

```
      DO 1 I=1,N
      J=2*I+LOCTM1
      IF(C(J)-X) 1,6,2
    1 CONTINUE
      GO TO 5
    2 IF(J.GT.(LOCTM1+2))GO TO 4
      J=3+LOCTM1
    5 CALL ERROR(LOCT)
    4 Y=C(J-1)+(C(J+1)-C(J-1))*(X-C(J-2))/(C(J)-C(J-2))
      RETURN
    6 Y=C(J+1)
      RETURN
      END
      SUBROUTINE TFFS1
C
C          ENGINE THRUST
C
C        INPUT REQUIRED (FROM DIRCOM)
C     ALT77F, AMACH, AMASS, AMASS1, AMASZS, AMT77F,
C     ANT77F, INDTFF, TXB7P, TYB7P, TZB7P, IN, N(5)
C     IT1.W, IT1.X, ZN(5), YN(5)
C
      COMMON/DIRCOM/DM1(126),ALT77F,DM2(6),AMACH,DM3(2),
     *AMASS,AMASS1,AMASZS,DM4(5),AMT77F,DM5(13),ANT77F,
     *DM6(495),INDTFF,DM7(246),TXB7P,DM8(2),TYB7P,DM9(2),
     *TZB7P,DM10(2),IT11W,IT11X,DM11(2),IT11W,IT11X,DM12(221),
     *AMASFS(2),AMASF1(2),DM13(444),IN,ZN(5),YN(5),N(5),DM14(290),
     *DM15(2.68)
C
C
C     IN     - NUMBER OF ENGINES
C     N(I)   - THROTTLE POSITION FOR I ENGINE
C     YN(I)  - ENGINE POSITIAN FOR I ENGINE
C     ZN(I)  - ENGINE POSITIAN FOR I ENGINE
C     MT(I)  - ENGINE PITCH MOMENT FOR I ENGINE
C     NT(I)  - ENGINE YAW MOMENT FOR I ENGINE
C     T(I)   - ACTUAL ENGINE THROUST FOR I ENGINE
C
      DIMENSION S(2)
      COMMON/FLXOP/DM1(300),T(5),BM2(303)
      COMMON/TABSPC/LOC(2),DUMM1(108)
      REAL N,MT(5),NT(5)
      DATA FMT1/6H  TFFS/,
     CS(1),S(2)/2HMT,2HNT/
C
C     INITIALIZATION BEFORE DATA READ IN
C
      TXB7P=0.
      TYB7P=0.
      TZB7P=0.
      ALT77F=0.
      AMT77F=0.
      ANT77F=0.
      RETURN
C
C     THRUST COMPUTATION SECTION
C
      ENTRY TFFS3
      TXB7P=0.
      TYB7P=0.
      TZB7P=0.
      ALT77F=0.
      AMT77F=0.
      ANT77F=0.
      IF(INDTFF.EQ.0)RETURN
```

```
      DO 10 I=1,IN
      CALL  HIRD13,LOC(1),IT10W,IT10X,DU,DU,N(I),AMACH,
     CDU,DU,T(I))
      MT(I)=T(I)*ZN(I)
      NT(I)=-T(I)*YN(I)
      TX87P=TX87P+T(I)
      AMT77F=AMT77F+MT(I)
      ANT77F=ANT77F+NT(I)
 10   CONTINUE
      RETURN
C
C     INITIAL PRINT
C
      ENTRY TFFS4
      RETURN
C
C     COMPUTE AND PRINT CODES
C
      ENTRY TFFS5
      IF(INDTFF.EQ.0)RETURN
      CALL STFL(3,1,FMT1)
      CALL STFL(2,2,S)
      RETURN
C
C     TIME HISTORY PRINT
C
      ENTRY TFFS6
      IF(INDTFF.EQ.0)RETURN
      CALL STFL(3,1,FMT1)
      CALL STOVAR(2,AMT77F,ANT77F,DU,DU,DU,DU,DU,DU)
      CALL STFL(0,1,DU)
      CALL LINES(2)
      WRITE(6,60) (T(I),I=1,IN)
 60   FORMAT(1H0,63X,4HT(I)/(14X,5E20.8))
      RETURN
C
C     UPDATE INTEGRATION
C
      ENTRY TFFS7
      RETURN
      END
      SUBROUTINE TFFS8(TC,TN)
C
C     SEARCH THROUGH RANGES OF N TO FIND THE N
C     THAT CORRESPONDS WITH THRUST TC AND CURRENT
C     MACH NUMBER
C
C     INPUT - TC (THRUST) AND MACH (CURRENT AMACH NO.)
C     OUTUT - TN (THROTTLE SETTING)
C
      COMMON/DIRCOM/DM1(226),ALT77F,DM2(6),AMACH,DM3(2),
     *AMASS,AMASS1,AMASZS,DM4(5),AMT77F,DM5(13),ANT77F,
     *DM6(495),INDTFF,DM7(246),TX87P,DM8(2),TY87P,DM9(2),
     *TZ87P,DM10(2),IT10W,IT10X,DM11(2),IT11W,IT11X,DM12(221),
     *AMASFS(2),AMASF1(2),DM13(444),IN,ZN(5),YN(5),N(5),DM14(290),
     *DM15(2068)
C
      COMMON/TABDIR/C(1)
      COMMON/TABSRC/LOC(2),DUMM1(108)
C
      IF(INDTFF.NE.0)GO TO 26
      TN=0.
      RETURN
 26   LOCT=LOC(1)-1
      DO 20 I=1,IT10W
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      K=LOCT+I
      CALL HIMO(3,LOC(1),IT10H,IT10X,DU,DU,C(K),AMACH,DU,DU,TH2)
      IF(C(K).GT.(-2.))GO TO 21
      IF(TC.GT.TH2)GO TO 19
      TN=-2.
      RETURN
21    IF(C(K).GT.(-1.))GO TO 22
      IF(TC.GT.TH2)GO TO 19
25    TN=C(K-1)+(TH1-TC)*(C(K)-C(K-1))/(TH1-TH2)
      RETURN
22    IF(C(K).EQ.1.)GO TO 27
      IF(C(K).GT.1.)GO TO 23
      IF(TC.GT.TH2)GO TO 19
      TN=-1.
      RETURN
27    IF(TC.GT.TH2)GO TO 19
      TN=1.
      RETURN
23    IF(C(K).GT.2.)GO TO 24
      IF(TC.GT.TH2)GO TO 19
      GO TO 25
24    TN=2.
      RETURN
19    TH1=TH2
20    CONTINUE
      RETURN
      END
      SUBROUTINE TFFS9(THRSET,THRUST)
C
C     COMPUTE THRUST AS A FUNCTION OF CURRENT MACH NO.,
C     AND DESIGNATED THROTTLE SETTING
C
C     INPUT  - THRSET(THROTTLE SETTING)
C     INPUT  - AMACH(MACH NO. FROM DIRCOM)
C     OUTPUT - THRUST
C
      COMMON/DIRCOM/DM1(126),ALT77F,DM2(6),AMACH,DM3(2),
     *AMASS,AMASS1,AMASZS,DM4(5),AMT77F,DM5(13),ANT77F,
     *DM6(495),INDTFF,DM7(246),TXB7P,DM8(2),TYB7P,DM9(2),
     *TZB7P,DM10(2),IT10H,IT10X,DM11(2),IT11W,IT11X,DM12(221),
     *AMASFS(2),AMASF1(2),DM13(444),IN,ZN(5),YN(5),N(5),DM14(290),
     *DM15(2068)
C
      COMMON/TABSRC/LOC(2),DUMM1(108)
C
      IF(INDTFF.NE.0)GO TO 30
      THRUST=0.
      RETURN
30    CALL HIMO(3,LOC(1),IT10H,IT10X,DU,DU,THRSET,AMACH,
     CDU,DU,THRUST)
      RETURN
      END
      SUBROUTINE VFCS1
C             VEHICLE PHYSICAL CHARACTERISTICS
      DIMENSION A(4)
      COMMON /DIRCOM/
     1DM(29)       ,
     2AIXXBS       ,AIXXS1     ,AIXYBS     ,AIXYS1     ,AIXZBS     ,AIXZS1     ;
     3DM1          ,AIYYBS     ,AIYYS1     ,AIYZBS     ,AIYZS1     ,DM2        ;
     4AIZZBS       ,AIZZS1     ,DM3(68)    ;
     5ALLJDF       ,ALMJDF     ,DM4(2)     ;            ALNJDF     ,DM5(11)    ,
     6ALYJDF       ,ALZJDF     ,DM6(7)     ;
     7AMASS        ,DM7(25)    ,
     8AREFF        ,DM8(187)   ;
     9DXCGF        ,DM9(5)     ,
```

```
     1D1RFF    ,DM1       ,D2RFF     ,DM11(19)  ,
     2EPS18    ,EPS19     ,DM12      ,EPS20     ,EPS21     ,EPS22     ,
     3EPS23    ,EPS24     ,DM13(215) ,
     4INDIOT   ,DM14(3)   ,                                INDJDP     ,
     5DM15(53) ,
     6INDISO   ,INDVPC    ,DM16(2)   ,           INDXYS    ,INDXZS    ,
     7DM17(223) ,
     8TIMES    ,DM18(74)  ,
     9XCGBF    ,XCGRF     ,DM19( 930),DM20(2068)
      COMMON/TABVPC/DUMM1(80),LOC(18),DUMM2(21)
      EQUIVALENCE (BS1IXX,AIXXS1),(BS1IYY,AIYYS1),(BS1IZZ,AIZZS1),
     *(BS1IXY,AIXYS1),(BS1IXZ,AIXZS1),(BS1IYZ,AIYZS1)
      DATA
     1(A(I),I=1,4)/5HXCGRF,5HAREFF,5HD1RFF,5HD2RFF/,
     2U,HDC1/5HAMASS,6H  VPCS/
    2 FORMAT (1H0,19X,28HINITIAL PRINT OUT FOR VPCS)
      RETURN
C     *********************************
C     INITIALIZATION AFTER DATA READ IN
      ENTRY VPCS2
      IF (INDVPC .EQ. 0) RETURN
C     *********************************
  173 BS1IXX=0.
      BS1IYY=0.
      BS1IZZ=0.
      BS1IXY=0.
      BS1IXZ=0.
      BS1IYZ=0.
      ALLJDF=0.
      ALMJDF=0.
      ALNJDF=0.
      ALYJDF=0.
      ALZJDF=0.
      IF (INDXZS.EQ.0) GO TO 212
      AIXYBS=0.
      AIYZBS=0.
  212 IF (INDXYS.NE.0) AIXZBS=0.
      RETURN
C     ***************************************************
C     VEHICLE PHYSICAL CHARACTERISTICS SUBPROGRAM SECTION
C     ***************************************************
      ENTRY VPCS3
      IF (INDVPC .EQ. 0) RETURN
C     ***************************************************
  216 IF (INDISO.EQ.0) GO TO 451
  223 CALL TLU(AMASS,LOC(1),XCGBF)
      CALL TLU(AMASS,LOC(2),AIXXBS)
      CALL TLU(AMASS,LOC(3),AIYYBS)
      CALL TLU(AMASS,LOC(4),AIZZBS)
      XCGBF=XCGBF+EPS18
      AIXXBS=AIXXBS+EPS19
      AIYYBS=AIYYBS+EPS20
      AIZZBS=AIZZBS+EPS21
      IF (INDXZS.NE.0) GO TO 335
      CALL TLU(AMASS,LOC(5),AIXYBS)
      CALL TLU(AMASS,LOC(7),AIYZBS)
      AIXYBS=AIXYBS+EPS22
      AIYZBS=AIYZBS+EPS24
  335 IF (INDXYS.NE.0) GO TO 353
      CALL TLU(AMASS,LOC(6),AIXZBS)
      AIXZBS=AIXZBS+EPS23
  353 IF (INDJDP.EQ.0) GO TO 451
      CALL TLU(XCGBF,LOC(8),ALYJDF)
      CALL TLU(XCGBF,LOC(9),ALZJDF)
      CALL TLU(XCGBF,LOC(10),ALLJDF)
```

```
      CALL TLU(XCGBF,LOC(11),ALMJDF)
      CALL TLU(XCGBF,LOC(12),ALNJDF)
451   IF (INDIUT.EQ.0) GO TO 567
      CALL TLU(TIMES,LOC(13),BS1IXX)
      CALL TLU(TIMES,LOC(14),BS1IYY)
      CALL TLU(TIMES,LOC(15),BS1IZZ)
563   IF (INDXZS.NE.0) GO TO 551
      CALL TLU(TIMES,LOC(16),BS1IXY)
      CALL TLU(TIMES,LOC(18),BS1IYZ)
551   IF (INDXYS.NE.0) GO TO 567
      CALL TLU(TIMES,LOC(17),BS1IXZ)
567   DXCGF=XCGBF-XCGRF
      RETURN
C     **************
C     INITIAL PRINT
      ENTRY VPCS4
      IF (INDVPC .EQ. 0) RETURN
C     **************
 27   CALL STFL(0,1,DU)
      WRITE (6,2)
      CALL STFL(2,4,A)
      CALL STOVAR(4,XCGRF,AREFF,D1RFF,D2RFF,DU,DU,DU,DU)
      RETURN
C     ************************
C     COMPUTE AND PRINT CODES
      ENTRY VPCS5
      IF (INDVPC .EQ. 0) RETURN
C     ************************
 64   CALL STFL(3,1,HBCI)
      CALL STFL(2,1,B)
      RETURN
C     ******************
C     TIME HISTORY PRINT
      ENTRY VPCS6
      IF (INDVPC .EQ. 0) RETURN
C     ******************
 103  CALL STFL(3,1,HBCI)
      CALL STFL(1,1,AMASS)
      RETURN
C     ***********
      ENTRY VPCS7
C     ***********
      RETURN
      END
      SUBROUTINE SACS1
      EXTERNAL ASIN
C              AERODYNAMIC FORCES AND MOMENTS
      LOGICAL INDAFC
      DIMENSION B(3),D(3),TMP(20),INC(79),TC(79)
      COMMON /DIRCOM/
     1DM(22)      ,
     2AA77P       ,DM1(81)    ,
     3ALA77F      ,DM2(3)     ,                ALIFTP     ,DM3(7)     ,
     4ALPHD,DM4(3),ALPHR1,DM5(3),
     5ALPTD       ,DM6(8)     ,
     6AMACH(3)    ,DM7(4)     ,                           AMA77F      ,
     7DM8(3),AMT77F,DM56(5),
     8ANAZ7F      ,ANA77P     ,DM9(6)     ,ANT77F,
     9ANUA7F      ,DM10(2)    ,                AREFF      ,DM11(11)   ,
     1BETAD,DM12(3),BETAR1,DM13(22),
     2CA          ,DM14(2)    ,                CAMNU      ,DM15(6)    ,
     3CD          ,DM16(2)    ,                CDMNU      ,DM17(6)    ,CL      ,
     4DM18(3)     ,                            CLMNU      ,CM         ,CMMNU   ,
     5CN2MNU,CN1MNU,DM19,CPHIA,DM20(9),
     6CRM         ,DM21(7)    ,
```

```
     7CY       ,DM22      ,CYM       ,CYMNU     ,DM23(49)           ,
     8DCU1     ,DCV1      ,DCW1      ,DCU2      ,DCV2      ,DCW2    ,
     9DCU3     ,DCV3      ,DCW3      ,DM24(15)
      COMMON /DIRCOM/
     1DELPD    ,DM25      ,DELQD     ,DM26      ,DELRD     ,DM27(17) ,
     2DRAGP    ,DM28      ,DXCGF     ,DM29(3)   ,
     3DYNPP    ,DM30      ,D1RFF     ,DM31      ,D2RFF     ,DM32(6)  ,
     4EPS1     ,EPS10     ,EPS11     ,EPS12     ,DM33(7)   ,
     5EPS2     ,DM34(7)   ,
     6EPS3     ,EPS4      ,EPS5      ,EPS6      ,EPS7      ,EPS8     ,
     7EPS9     ,DM35(43),HGC7F,DM61(44),
     5INDAER   ,DM37(2)   ,          INDARC     ,DM38(3)
      COMMON /DIRCOM/
     1INDA01   ,INDA02    ,INDA03    ,INDA04    ,INDA05    ,INDA06   ,
     2INDA07   ,INDA08    ,INDA09    ,INDA10    ,INDA11    ,INDA12   ,
     3INDA13   ,INDA14    ,INDA15    ,INDA16    ,INDA17    ,INDA18   ,
     4INDA19   ,INDA20    ,INDA21    ,INDA22    ,INDA23    ,INDA24   ,
     4INDA25   ,INDA26    ,INDA27    ,INDA28    ,INDA29    ,INDA30   ,
     5INDA31   ,INDA32    ,INDA33    ,INDA34    ,INDA35    ,INDA36   ,
     6INDA37   ,INDA38    ,INDA39    ,INDA40    ,INDA41    ,INDA42   ,
     7INDA43   ,INDA44    ,INDA45    ,INDA46    ,INDA47    ,INDA48   ,
     8INDA49   ,INDA50    ,INDA51    ,INDA52    ,INDA53    ,INDA54   ,
     9INDA55   ,INDA56    ,INDA57    ,INDA58    ,INDA59    ,INDA60   ,
     1INDA61   ,INDA62    ,INDA63    ,INDA64    ,INDA65    ,INDA66   ,
     2INDA67   ,INDA68    ,INDA69    ,INDA70    ,INDA71    ,INDA72   ,
     3INDA73   ,INDA74    ,INDA75    ,INDA76    ,INDA77    ,INDA78   ,
     4INDA80   ,DM52(13)   ,INDBAD    ,DM39(7)
      COMMON /DIRCOM/
     7INDFLT    ,DM40(55)   ,
     8INDSDF    ,DM41(23)   ,
     8NSYMD     ,DM42(140)  ,
     9SIDEP     ,DM43(11)   ,
     9SPHIA     ,DM44(111)  ,
     1VA77F,DM45(11),
     2VS77F     ,DM46(35)   ,
     4YA77P     ,DM47(8),ZGW7F1,DM59(179),
     5PI77R,DM48(7),
     6QI77R,DM49(7),
     7RI77P,DM50(13),TS77R,DM51(420),IAP,DM57(3),
     *ALPHDS,ALPHDL,FCG,DM53(94),DELQN,DM54(7),
     *DELQL,DELQU,DM55(7),DELRL,DELRU,DM60(78),HP,DM58(64),DDM77(2068)
      COMMON/TABSRC/DUMM1(2),LOC(79),DUMM2(29)
C
      COMMON/AUTSAC/ALPDR1,QDESP,CLRR,ALPDES,CDR,DELRN
C
      COMMON/LG/FXM,FYM,FZM,LM,MM,NM,EPSLOL
      REAL LM,MM,NM
C
      EQUIVALENCE (INDA01,INC(1)),
     *(TC( 1),CAALPH),(TC( 2),CAALSQ),(TC( 3),CABETA),(TC( 4),CABTSQ),
     *(TC( 5),CADLTQ),(TC( 6),CADLSQ),(TC( 7),CAALDT),(TC( 8),CAALDL),
     *(TC( 9),CABTDL),(TC(10),CN1ZER),(TC(11),CN1ALP),(TC(12),CN1ASQ),
     *(TC(13),CN1BLT),(TC(14),CN1BSQ),(TC(15),CN1DLQ),(TC(16),CN1DSQ),
     *(TC(17),CN1ALB),(TC(18),CN1ALD),(TC(19),CN1BDL),(TC(20),CN1ADL),
     *(TC(21),CN1ADX),(TC(22),CN1Q  ),(TC(23),CN1QX ),(TC(24),CYZERO),
     *(TC(25),CYALPH),(TC(26),CYALSQ),(TC(27),CYBETA),(TC(28),CYBTSQ),
     *(TC(29),CYDLLR),(TC(30),CYDRSQ),(TC(31),CYALBT),(TC(32),CYALDL),
     *(TC(33),CYBTDL),(TC(34),CYBTDT),(TC(35),CYBTDX),(TC(36),CYR   ),
     *(TC(37),CYRX  ),(TC(38),CLZERO),(TC(39),CLALPH),(TC(40),CLALSQ),
     *(TC(41),CLBETA),(TC(42),CLBTSQ),(TC(43),CLDLTP),(TC(44),CLDLSQ),
     *(TC(45),CLALPT),(TC(46),CLALDL),(TC(47),CLBTDL),(TC(48),CLP   ),
     *(TC(49),CLR   ),(TC(50),CLRX  ),(TC(51),CMZERO),(TC(52),CMALPH),
     *(TC(53),CMALSQ),(TC(54),CMBETA),(TC(55),CMBTSQ),(TC(56),CMDLTQ),
     *(TC(57),CMDLSQ),(TC(58),CMALBT),(TC(59),CMALDL),(TC(60),CMBTDL),
     *(TC(61),CMALDT),(TC(62),CMADTX),(TC(63),CMQ   ),(TC(64),CMQX  ),
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
     *(TC(65),CN2ZER),(TC(66),CN2ALP),(TC(67),CN2ASQ),(TC(68),CN2BET),
     *(TC(69),CN2BSQ),(TC(70),CN2DLT),(TC(71),CN2DSQ),(TC(72),CN2ALB)
      EQUIVALENCE
     *(TC(73),CN2ALD),(TC(74),CN2BDL),(TC(75),CN2BTD),(TC(76),CN2BDX),
     *(TC(77),CN2R  ),(TC(78),CN2RX ),(TC(79),CAVAH)
      EQUIVALENCE (CY,CYA),(DYNPL,DYNPP)
      DATA DEGRAD/.01745329/
      DATA
     *HBCI,C/6H  SAC1,6HCAVAH /,
     1(B(I),I=1,3)/2HCA,2HCN,2HCY/,
     2(D(I),I=1,3)/2HCL,2HCM,3HCNN/
1     FORMAT (1H0,15X,26HINITIAL PRINT OUT FOR SAC1)
1267  ANA77P=0.
      ALA77F=0.
      AMA77F=0.
      ANAZ7F=0.
      AA77P=0.
      YA77P=0.
      RETURN
C     ***********
      ENTRY SACS3
C     ***********
      IF (INDAER .EQ. 0) RETURN
      IF (INDARC        ) GO TO 1267
      IF(VS77F.NE.0.)GO TO 1470
      CA =CAMNU
      CN =CN1MNU
      CY =CYMNU
      CRM=CLMNU
      CM =CMMNU
      CYM=CN2MNU
1401  TMP(1)=EPS5*CY+EPS6
      TMP(2)=DYNPP*AREFF
      YA77P=TMP(1)*TMP(2)
      ALA77F=(EPS7*CRM+EPS8)*TMP(2)*D2RFF
      ANAZ7F=(EPS11*CYM+EPS12)*TMP(2)*D2RFF
      TMP(1)=EPS3*CA+EPS4
      TMP(2)=DYNPP*AREFF
      AA77P=TMP(1)*TMP(2)
      ANA77P=(EPS1*CN+EPS2)*TMP(2)
      AMA77F=(EPS9*CM+EPS10)*TMP(2)*D1RFF
      RETURN
1470  IGO1=1
      IGO2=1
      IGO3=1
      DO 71 I=1,79
71    TC(I)=0.
50    HG=HGC7F
      IF(IAP.GE.3)HG=HR
C   EPSLO2 IS COMPUTED IN SUBROUTINE LGEA3C
      TMP(1)=EXP(-4.E*(HG-EPSLO2-HCG)/(D2RFF-HCG))  ,
      DO 79 I=1,2
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
79    CONTINUE
      DO 80 I=15,16
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
80    CONTINUE
      DO 81 I=10,12
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
81    CONTINUE
      DO 85 I=20,22,2
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
85    CONTINUE
      IF(INDAB0.NE.0)CAVAH =(AERO1(LOC(79),AERO2)-AERO2)*TMP(1)+AERO2
      GO TO (10,20,30),IGO1
```

```
1     DO 82 I=51,53
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
82    CONTINUE
      DO 83 I=59,63
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
83    CONTINUE
      GO TO (11,40),IGO2
11    DO 84 I=27,26
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
84    CONTINUE
      DO 70 I=61,71
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
70    CONTINUE
      GO TO (12,60),IGO3
12    DO 72 I=3,9
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
72    CONTINUE
      DO 73 I=13,14
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
73    CONTINUE
      DO 74 I=17,28
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
74    CONTINUE
      DO 75 I=30,50
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
75    CONTINUE
      DO 76 I=54,55
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
76    CONTINUE
      DO 77 I=64,67
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
77    CONTINUE
      DO 78 I=72,78
      IF(INC(I).NE.0)TC(I)=(AERO1(LOC(I),AERO2)-AERO2)*TMP(1)+AERO2
78    CONTINUE
      TMP(2 )=ABS(ALPHD)
      TMP(4 )=ABS(BETAD)
      TMP(3 )=BETAD**2
      TMP(5 )=ABS(DELQD)
      TMP(6)=ALPHD*ALPHD
      TMP(7)=ABS(DELRD)
      TMP(8)=.5*DREFF/VA77F
      CD =CAVAR+TMP(2)*CAALPH+ALPHD**2*CAALSQ+TMP(4)*CABETA+TMP(3)*
     *     CABTSQ+TMP(5)*CADLTQ+DELQD**2*CADLSQ+ABS(ALPHD*DELQD)*CAALDL+
     *     ABS(ALPHD*BETAD)*CAALBT+ABS(BETAD*DELQD)*CABTDL
      CL =CNZERO+ALPHD*CN1ALP+ALPHD*TMP(2)*CN1ASQ+TMP(4)*CN1BET+
     *     TMP(3)*CN1BSQ+DELQD*CN1DLQ+TMP(5)*DELQD*CN1DSQ+TMP(2)*DELQD*
     *     CN1ALD+TMP(4)*ALPHD*CN1ALB+TMP(4)*DELQD*CN1BDL+
     *     D1RFF/VA77F*ALPHP1/2.*(CN1ADX*DXCGF+CN1ADL)+
     *     D1RFF/VA77F*QI77P/2.*(CN1QX  *DXCGF+CN1Q   )
      TMP(9)=ALPHD*.01745329
      CA=CD*COS(TMP(9))-CL*SIN(TMP(9))
      CN=CL*COS(TMP(9))+CD*SIN(TMP(9))
      CY=CYZERO+TMP(2)*CYALPH+TMP( 6)*CYALSQ+BETAD*CYBETA+BETAD*TMP(4)*
     *     CYBTSQ+DELRD *CYDELR+TMP( 7)*DELRD*CYDRSQ+TMP(2)*DELRD*CYALDL+
     *     TMP(2)*BETAD *CYALBT+TMP( 4)*DELRD*CYBTDL+
     *     BETAR1*TMP(8)*(CYBTDX*DXCGF+CYBTDT)+
     *     RI77R *TMP(8)*(CYRX  *DXCGF+CYR    )
      TMP(8 )=ABS(ALPHD)
      TMP(10)=ALPHD**2
      TMP(9 )=ABS(BETAD)
      TMP(10)=ABS(DELRD)
      TMP(2 )=D2RFF/(VA77F*2.)
      TMP(3)=BETAD*BETAD
      TMP(4)=ABS(DELQD)
```

```
      CRM=CLZERO+TMP(8)*CLALPH+TMP(11)*CLALSQ+TMP(9)*CLBETA+BETAD**2*
     *    CLBTSQ+DELPC *CLDLTP+ABS(DELPD)*DELPD*CLDLSQ+TMP(8)*DELPD*
     *    CLALDL+ABS(ALPHD*BETAD)*CLALBT+TMP(9)*DELPD*CLBTDL+
     *    TMP(2)*PI77R*CLP+PI77R*TMP(2)*(CLRX*DXCGF+CLR)
      CM =CMZERO+ALPHD*CMALPH+TMP(8)*ALPHD*CMALSQ+TMP(9)*CMBETA+TMP(3)*
     *     CMBTSQ+DELQD*CMDLTQ+TMP(4)*DELQD*CMDLSQ+TMP(8)*CMALDL*DELQD +
     *     TMP(9)*ALPHD*CMALBT+TMP(9)*DELQD*CMBTDL+
     *     D1RFF/VA77F*ALPHR1/2.*(CMADTX*DXCGF+CMALDT)+
     *     D1RFF/VA77F*QI77R /2.*(CMQX   *DXCGF+CMQ    )-
     *     DXCGF/D1RFF*CN
      CYM=CN2ZER+TMP(8)*CN2ALP+TMP(11)*CN2ASQ+BETAD*CN2BET+TMP(9)*BETAD*
     *     CN2BSQ+DELRD *CN2DLT+TMP(10)*CN2DSQ*DELRD+TMP(8)*DELRD*CN2ALD+
     *     TMP(8)*BETAD *CN2ALB+TMP(9 )*CN2BDL*DELRD+
     *     TMP(2)*BETAR1*(CN2BDX*DXCGF +CN2BTD)+
     *     TMP(2)*RI77R *(CN2RX *DXCGF +CN2R  )-CY*DXCGF/D2RFF
      GO TO 14.1
C     ***********
      ENTRY SACS4
C     ***********
      IF (INDAER .EQ. 0) RETURN
      RETURN
C     ***********
      ENTRY SACS5
C     ***********
      IF (INDAER .EQ. 0) RETURN
 63   CALL STFL(3,1,HBCI)
      IF(INDA80.NE.0)CALL STFL(2,1,C)
      CALL STFL(2,3,0)
 123  CALL STFL(2,3,C)
      RETURN
C     ***********
      ENTRY SACS6
C     ***********
      IF (INDAER .EQ. 0) RETURN
 326  CALL STFL(3,1,HBCI)
      IF(INDA80.NE.0)CALL STFL(1,1,CAVAH)
      CALL STOVAR(3,CA,CN,CY,DU,DU,DU,DU,DU)
 370  CALL STOVAR(3,CRM,CM,CYM,DU,DU,DU,DU,DU)
      RETURN
C     ***********
      ENTRY SACS7
C     ***********
      RETURN
C
C     DETERMINE THE ALPHA THAT MAKES CL=CLRR AND THE
C         ASSOCIATED CDR
C
      ENTRY SACS8
C
C         INPUT FROM CALLING PROGRAM
C     CLRR
C         OUTPUT
C     ALPDES, CDR
C
C         INPUT FROM DIRCOM
C     DELQN, ALPHDS, ALPHDL, HCGD2RFF, INDA01, INDA02, HR,
C     INDA15, INDA16, INDA38, INDA39, INDA40, INDA80
C     INDA20, INDA22, D1RFF, VA77F, PI77F, ALPHR1
C
      IF((IAP.LT.3).OR.(ABS(HG-HR).LE.1.E-6))GO TO 20
      IGO1=2
      GO TO 50
 20   TMP(2)=CN1ZER+CN1DLQ*DELQN+CN1DSQ*ABS(DELQN)*DELQN-CLRR
     C+.5*CN1Q*QDESR*D1RFF/VA77F
      ALPDES=AQUAD(CN1ASQ,CN1ALP,TMP(2),ALPHDS,ALPHDL)
```

```
      COR=CAVAH+CAALPH*ABS(ALPDES)+CAALSQ*ALPDES*ALPDES
      RETURN
C
C     DETERMINE THE CL1 AND CD1 FOR ALPHA=A1
C
      ENTRY SACS9
C
C          INPUT FROM CALLING PROGRAM
C     ALPDES
C          OUTPUT
C     CLRR,COR
C          INPUT FROM DIRCOM
C     DELON, HCG, D2RFF, INDA01, INDA02, INDA15, INDA16, HR,
C     INDA38, INDA39, INDA40, INDA80,INDA20, INDA22,
C     D1RFF, VA77F, QI77R, ALPHR1
C
      IF((IAP.LT.3).OR.(ABS(HG-HR).LE.1.E-6))GO TO 30
      IGO1=3
      GO TO 50
   30 CLRR=CN1ZER+CN1ALP*ALPDES+CN1ASQ*ABS(ALPDES)*ALPDES+CN1DLQ
     C*DELON+CN1DSQ*ABS(DELON)*DELON
     C+.5*CN1Q*QDESR*D1RFF/VA77F
      COR=CAVAH+CAALPH*ABS(ALPDES)+CAALSQ*ALPDES*ALPDES
      RETURN
C
C     EVALUATE NOMINAL DELON
C
      ENTRY SACS10
C
C          INPUT FROM CALLING PROGRAM - ALPDES
C          OUTPUT - DELON
C          INPUT FROM DIRCOM
C     DELQL, DELQU, HCG, D1RFF, D2RFF, DYNPP, AREFF, DXCGF, HR,
C     AMT77F, INDA01, INDA02, INDA15, INDA16, INDA38, INDA39, INDA40,
C     INDA80, INDA51, INDA52, INDA53, INDA56, INDA57
C     INDA61, INDA63, VA77F, ALPHR1, QI77R
C
      IF((IAP.LT.3).OR.(ABS(HG-HR).LE.1.E-6))GO TO 40
      IGO1=1
      IGO2=2
      GO TO 50
   40 TMP(2)=CN1ZER+ALPDES*(CN1ALP+CN1ASQ*ABS(ALPDES))
     C+CN1DLQ*DELQN+CN1DSQ*ABS(DELQN)*DELQN
     C+.5*CN1Q*QDESR*D1RFF/VA77F
      TMP(3)=CAVAH+CAALPH*ABS(ALPDES)+CAALSQ*ALPDES*ALPDES
      TMP(4)=TMP(2)*COS(ALPDES*DEGRAD)+TMP(3)*SIN(ALPDES*DEGRAD)
      TMP(5)=DYNPP*AREFF*D1RFF
      TMP(2)=-TMP(4)*DXCGF*TMP(5)/D1RFF
      TMP(3)=CMDLTQ*TMP(5)
      TMP(4)=(CMZERO+ALPDES*(CMALPH+CMALSQ*ABS(ALPDES)))
     C*TMP(5)+AMT77F+TMP(2)+MM
     C+.5*CMQ*QDESR*D1RFF*TMP(5)/VA77F
      TMP(2)=TMP(5)*CMDLSQ
      DELQN=AQUAD(TMP(2),TMP(3),TMP(4),DELQL,DELQU)
      RETURN
C
C     EVALUATE NOMINAL DELRN
C
      ENTRY SACS11
C
C          INPUT FROM DIRCOM
C     IAP, D2RFF, DYNPP, AREFF, ANT77F, YGW7F1, VA77F
C     DELRL, DELRU
C
      IF((IAP.LT.3).OR.(ABS(HG-HR).LE.1.E-6))GO TO 60
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IGOJ=2
      GO TO 11
60    BETADE=0.
      IF(IAP.GE.3)BETADE=ASIN(-YGW7F1/VA77F)*57.2957795
      TMP(5)=DYNPP*AREFF*DZREFF
      TMP(6)=DXCGF/DZREF
      TMP(3)=CN2DLT*TMP(5)
      TMP(2)=CN2DSQ*TMP(5)
      TMP(4)=ANT77F+BETADE*(CN2BET-CYBETA*TMP(6)
     C+ABS(BETADE)*(CN2BSQ-CYBTSQ*TMP(6)))*TMP(5)
      DELRN=AQUAD(TMP(2),TMP(3),TMP(4),DELRL,DELRU)
      RETURN
      END
      FUNCTION AERO1(LOCT,AERO2)
      COMMON/TABDIP/C(1)
      AERO1=C(LOCT)
      AERO2=C(LOCT+1)
      RETURN
      END
      FUNCTION AQUAD(A,B,C,XLLIM,XULIM)
      IF(ABS(A).GT.1.E-10)GO TO 4
      X=0.
      IF(ABS(B).LE.1.E-10)GO TO 11
      X=-C/B
      GO TO 11
4     TMP=B*B-4.*A*C
      IF(TMP.LT.0.)GO TO 30
      TMP1=SQRT(TMP)
      X=(-B+TMP1)*.5/A
      IF(X.LT.0.)GO TO 5
      IF((X.LT.XULIM).AND.(X.GT.XLLIM))GO TO 20
5     X=(-B-TMP1)*.5/A
      IF(X.LT.0.)GO TO 30
      IF((X.LT.XULIM).AND.(X.GT.XLLIM))GO TO 20
30    TMP=B*B+4.*A*C
      IF(TMP.LT.0.)GO TO 10
      TMP1=SQRT(TMP)
      X=(-B+TMP1)*.5/(-A)
      IF(X.GT.0.)GO TO 15
      IF((X.LT.XULIM).AND.(X.GT.XLLIM))GO TO 20
15    X=(-B-TMP1)*.5/(-A)
      IF(X.GT.0.)GO TO 10
11    IF((X.LT.XULIM).AND.(X.GT.XLLIM))GO TO 20
10    FX1=XLLIM*(A*ABS(XLLIM)+B)+C
      FX2=XULIM*(A*ABS(XULIM)+B)+C
      X=XLLIM
      IF(ABS(FX1).GT.ABS(FX2))X=XULIM
20    AQUAD=X
      RETURN
      END
      SUBROUTINE OPT1
      EXTERNAL ASIN,ATAN2
C               SIX-DEGREE-OF-FREEDOM OVER FLAT PLANET
      DIMENSION
     *         TMP(27),B(17),D(2),E(2),DRAGC(4),
     *      F        ( 1),H        ( 2),J        ( 3),K        ( 3),
     *L(3),G(2),LA(19)
      LOGICAL INDARC , SW
      COMMON /DIRCOM/
     *DM(22),AA77P,AE77F,DM2(4),AIXR7S,AIXXBS,
     *AIXXS1,AIXYBS,AIXYS1,AIXZBS,AIXZS1,DM3,
     *AIYYBS,AIYYS1,AIYZBS,AIYZS1,DM4,AIZZBS,
     *AIZZS1,DM5(7),AKLT,AKL1,AKL2,AKL3,
     *      AKL4        ,AKL5        ,AKL6        ,AKMT        ,AKM1        ;
     *      AKM2        ,AKM3        ,AKM4        ,AKM5        ,AKM6        ;
```

```
*        AKNT           ,AKN1           ,AKN2           ,AKN3           ,AKN4           *
*AKN5,AKN6,DM6(31),ALAMTD,ALAMTR,ALA77F,
*ALB77F,DM7(5),ALLJDF,ALMJDF,DM8(2),ALNJDF,ALPHD,DM9(2),
*ALPHR,ALPH-1,ALPL1,DM10(2),ALPTD,ALPTR,
*ALT77F,ALYJDF,ALZJDF,DM11(4),AMACH,DM12(2),AMASS,
*AMASS1,DM13,AMAXH,AMA77F,AMB77F,DM14(2),AM177F,DM15(4),ANAD7G,
*ANAZ7F,ANA77F,ANB77F,DM16,ANGAMG,ANPSIG,
*ANSIGG,ANTHTG,ANT77F,DM17(2),ANVW7G,DM18,AXP7F,
*AX77F,AYP7F,AY77F,AZP7F,DM19,AZ77F,DM20(4),
*BETAD,DM21(2),BETAR,BETAP1,BETL1,DM22(37),CGAMA,
*CGAMD,DM23(4),CLAMT,DM24(7),CPHIA,CPHIB,DM25,CPHIT,DM26(2),
*CPSID,DM27(4),CSIGA,CSIGD,DM28(2),CTHTD,DM29,CTHTR,DM30(35)
 COMMON /DI-COM/
*        DCL1           ,DCM1           ,DCN1           ,DCL2           ,DCM2           *
*DCN2,DCL3,DCM3,DCN3,DM31(9),DCD1,DM32(30),
*DELT1,DELT2,DM33,DLFXP,DLFYP,DLFZP,DM34,
*DLLTF,DM35,DLMTF,DLNTF,DLTM1,DLTM2,DM36(2),
*DRAGP,DM37,DXCGF,DM38(3),DYNPP,DM39(37),FXB7P,DM40(3),FYB7P,
*DM41(3),FZB7P,DM42(4),GAMDD,GAMDP,DM199(5),GAML1,GAM7D,
*GAM7R,GAM7 1,GREFF,DM43,GXB7F,DM44(4),GYB7F,DM45(3),
*GZB7F,DM46(5),HGG7F,DM47(41),INDACM,INDADD,INDALP,DM48(2),
*INDAPG,INDARG,DM49(95),INDBAD,DM50(8),
*INDFPA,DM51,INDFPR,INDGCR,DM52,
*INDGRT,INDGSF,DM53(9),INDLDF,DM54(2),INDPLA,DM55(2),INDPSK,
*DM56,INDRMC,DM57(34),INDSDF,INDSKP,INDSPL,INDSTE,DM58,INDSTG,
*DM59(6),INDTSD,DM60,INDWGT,INDWIN,INDXYS,INDXZS,DM61(10),
*DMGRR,DMGRT,DM62(4),PA77P,DM63(3),PHIAD,PHIAR,PHIBD,
*DM64(10),PHIFD,PHIPR,DM65(2),PHITD,PHITR,DM66(7),
*PI77D,DM67(5),PSIBD,DM68(4),PSIPD,PSIPR,DM69(6),QI77D,DM70(43),
*PG77N,DM71(2),RHOAS,DM72(2),RI77D,DM73(28),SGAMA,SGAMD,
*DM74(4),SIGDD,SIGDR,SIGL1,DM75,SIG7D,SIG7R,
*SIG7R1,DM76,SLAMT,SPHIA,SPHIB,DM77,SPHIT,DM78(2)
 COMMON /DIPCOM/
*SPSID,DM79(2),SSIGA,SSIGD,DM80(13),STHTD,DM81,STHTR,DM82(17),
*T-77P,DM83(6),THTBD,DM84(3),THTPD,THTPR,THTPD,DM85(3),
*TIME,DM86(2),TIMES,DM87(3),TIMSX,DM198(2),TL11,TL12,DM88(6),TVACP,
1DM89,
*TXB7P,DM90(2),TYB7P,DM91(2),TZB7P,DM92(22),UW77F,DM93,
*VA77F,DM94(2),VGPRF,VG77F,DM95(7),VS77F,VW77F,DM96(9),WTR7P,
*WW77F,DM97(9),XGW7F1,DM98,XGZ7F,DM99(5),XN77F,DM100(5),YA77P,
*DM101(8),YGW7F1,DM102,YGZ7F,DM103(6),YN77F,DM104(9),ZGW7F1,
*DM105(7),ZN77F,DM106(108),AL1,DM107,AL17S1,DM108,AL2,
*DM109,AL27S1,DM110,AL3,DM111,AL37S1,DM112(5),AM1,DM113,
*AM17S1,DM114,AM2,DM115,AM27S1,DM116,AM3,DM117,AM37S1,
*DM118,AN1,DM119,AN177S,DM120,AN2,DM121,AN277S,DM122,AN3,
*DM123,AN377S,DM124,PHIDR,DM125(3),PI77R,DM126,
*PI77R1,DM127,PSIBP,DM128(3),QI77R,DM129,QI77R1,DM130,
*PD77N,DM131,PD77N1,DM132,RI77R,DM133,RI77R1,DM134(3),
*THTBP,DM135(11),U777F,DM136,U777F1,DM137,VD77F,DM138
 COMMON/DIPCOP/
*VD77F1,DM139,VGRVF,DM140,VGRVF1,DM141,VP77F,DM142,
*VP77F1,DM143,VTHLF,DM144,VTHLF1,DM145,V777F,DM146,
*V777F1,DM147,W777F,DM148,W777F1,DM149(7),XG77F,
*DM150,XG77F1,DM151(11),YG77F,DM152,YG77F1,DM153(11),
*ZG77F,DM154,ZG77F1,DM1(333),
*ICS,CDCH,SSH,XCH,YCH,ZCH,DM155(235),RWHGP,DM156(48),DM157(2068)
 COMMON/TAJ[illegible]FG/DUMM1(39),LOC(4) DUMM2(7)
 COMMON/LG/FXM,FYM,FZM,LM,MM,NM
 COMMON/FLXOP/BH(300).FCX,FCY,FCZ,BMH(300)
 REAL LM,MM,NM,MDX,MDY,MDZ
 EQUIVALENCE ( DM (6) , SW )
 DATA
1(B(I),I=1,17)/4HTIME,5HTIMES,5HXG77F,5HYG77F,5HHG77F,5HU777F,
25HV777F,5HW777F,5HPI77R,5HQI77R,5HRI77R,5HAMACH,5HVA77F,
35HDYNPP,6HXG77F1,6HYG77F1,6HZG77F1/,
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      4D(1),D(2)/5HALPHD,5HBETAD/,E(1),E(2)/6HALPHD1,6HBETAD1/,
      5F(1)/5HVG77F/,G(1),G(2)/5HGAM7D,5HSIG7D/,
      6H(1),H(2)/6HGAM7R1,6HSIG7R1/,(J(I),I=1,3)/5HTHTPD,5HPSIPD,
      75HPHIPD/,(K(I),I=1,3)/5HAXP7F,5HAYP7F,5HAZP7F/,
      8(L(I),I=1,3)/5HAX77F,5HAY77F,5HAZ77F/,
      *                P /6HTHTRD /
       DATA
      *      HBCI  ,  C   ,  M    /
      *     6H  2SOF,5HRG77N,5HWTR7P/,
      *(DRAGC(I),I=1,4)/3HFDC,3HFCX,3HFCY,3HFCZ/
 1     FORMAT (1H0,15X,36HPRINT CODES IDENTIFYING TIME HISTORY)
 2     FORMAT (1H ,15X,26HINITIAL PRINT OUT FOR 2SOF)
C45    PRE DATA INITIALIZATION
 45    INDSDF=2
       RETURN
C      **********
       ENTRY OPT2
C      **********
C                 (AL17S1,AL1   ,AL27S1,AL2   ,AL37S1,AL3    ,AM17S1,AM1 ;
C     *            AM27S1,AM2   ,AM37S1,AM3   ,AN177S,AN1    ,AN277S,AN2 ;
C     *            AN377S,AN3   ,U777F1,U777F ,V777F1,V777F  ,W777F1,
C     *            W777F ,PI77R1,PI77R ,QI77R1,QI77R ,RI77R1,RI77R ,
C     *            XG77F1,XG77F ,YG77F1,YG77F ,ZG77F1,ZG77F )
 62    CALL INUPD (18,LA)
       CALL LGEAR4
       CALL VPCS4
       CALL TFFS4
       CALL SACS4
       CALL FLEX4
       CALL STFL(2,1,DU)
       CALL LINES (2)
       WRITE (6,1)
       CALL STFL(3,1,HBCI)
       CALL STFL(2,17,B)
       IF(INDGCR.NE.0)CALL  STFL(2,1,C)
       IF(INDAPC.NE.0)CALL  STFL(2,2,D)
       IF(INDADD.NE.0)CALL  STFL(2,2,E)
       IF(INDGSR.NE.0)CALL  STFL(2,1,F)
 236   IF (INDFPA.EQ.0) GO  TO 250
       CALL STFL(2,2,G)
       IF(INDFPR.NE.0)CALL  STFL(2,2,H)
 250   IF(INDGRT.NE.0)CALL  STFL(2,3,J)
       IF(INDACM.EQ.1)CALL  STFL(2,3,K)
       IF(INDACM.EQ.2)CALL  STFL(2,3,L)
       IF(INDWGT.NE.0)CALL  STFL(2,1,M)
       IF (INDRMC.EQ.0) GO  TO 335
       CALL STFL(2,1,P)
C                                  (THTRR1,THTRR )
       CALL INUPD  (1,LTHTRR)
 335   CALL STFL(2,4,DRAGC)
       CALL LGEAR5
       CALL VPCS5
       CALL TFFS5
       CALL SACS5
       CALL FLEX5
       CALL STFL(2,1,DU)
       CALL DEF
C367   2SOF1
 367   ZG77F  =-HGC7F
       SIGDD  = SIG7D
       GAMDD  = GAM7D
       SIGDR  = SIGDD*.01745329
       GAMDR  = GAMDD*.01745329
       SIGL1  = SIGDR
       GAML1  = GAMDR
```

```
      SGAMD = SIN(GAMDR)
      CGAMD = COS(GAMDR)
      SSIGD = SIN(SIGDR)
      CSIGD = COS(SIGDR)
      THTBR = THTBD*.01745329
      PSIBR = PSIBD*.01745329
      PHIBR = PHIBD*.01745329
      SPHIB = SIN(PHIBR)
      CPHIB = COS(PHIBR)
      STHTB = SIN(THTBR)
      CTHTB = COS(THTBR)
      SPSIB = SIN(PSIBR)
      CPSIB = COS(PSIBR)
      DCL1=CPSIB*CTHTB
      DCL2=SPSIB
      DCL3=-CPSIB*STHTB
      DCM1=SPHIB*STHTB-CPHIB*SPSIB*CTHTB
      DCM2=CPHIB*CPSIB
      DCM3=SPHIB*CTHTB+CPHIB*STHTB*SPSIB
      DCN1=CPHIB*STHTB+CTHTB*SPSIB*SPHIB
      DCN2=-SPHIB*CPSIB
      DCN3=CPHIB*CTHTB-SPHIB*STHTB*SPSIB
      AL1    = DCL1
      AL2    = DCL2
      AL3    = DCL3
      AM1    = DCM1
      AM2    = DCM2
      AM3    = DCM3
      AN1    = DCN1
      AN2    = DCN2
      AN3    = DCN3
      XG77F1= CSIGD*CGAMD*VG77F
      YG77F1= SSIGD*CGAMD*VG77F
      ZG77F1=-SGAMD*      VG77F
      TMP(1)= XG77F1
      TMP(2)= YG77F1
      TMP(3)= ZG77F1
      CALL INTEG  (LA(16) , XG77F1 )
      CALL INTEG  (LA(17) , YG77F1 )
      CALL INTEG  (LA(18) , ZG77F1 )
      CALL MULT31(DCL1,TMP(1),TMP(4))
      U777F = TMP(4)
      V777F = TMP(5)
      W777F = TMP(6)
      INOSKP=1
      GO TO 702
C     ***********
      ENTRY OPT4
C     ***********
C621  SOF
  621 DCL1   = AL1
      DCL2   = AL2
      DCL3   = AL3
      DCM1   = AM1
      DCM2   = AM2
      DCM3   = AM3
      DCN1   = AN1
      DCN2   = AN2
      DCN3   = AN3
      TMP(1)= U777F
      TMP(2)= V777F
      TMP(3)= W777F
      CALL TRNF03  (DCL1,TMP(4))
      CALL MULT31  (TMP(4),TMP(1),TMP(13))
      XG77F1= TMP(13)
```

```
      YG77F1= TMP(14)
      ZG77F1= TMP(15)
      CALL INTEG (LA(16) , XG77F1 )
      CALL INTEG (LA(17) , YG77F1 )
      CALL INTEG (LA(18) , ZG77F1 )
C762  SDF2
 762  GXB7F = DCL3*GREFF
      GYB7F = DCM3*GREFF
      GZB7F = DCN3*GREFF
      HGC7F =-ZG77F
      CALL ATMS (HGC7F+RWHGR)
      IF (INDGCR.NE.0) RG77N =
     *                 SQRT((XG77F-XGZ7F)**2+(YG77F-YGZ7F)**2)*1.6457925E-4
      IF (INDWIN.NE.0) GO TO 754
      UW77F = 0.
      VW77F = 0.
      WW77F = 0.
      GO TO 1035
 754  GO TO (755,757,761),INDWIN
 755  TMP(1)=TIME
      GO TO 762
 757  TMP(1)=HGC7F
      GO TO 762
 761  TMP(1)=RG77N
 762  CALL TLU(TMP(1),LOC(2),XGW7F1)
      CALL TLU(TMP(1),LOC(3),YGW7F1)
      CALL TLU(TMP(1),LOC(4),ZGW7F1)
      TMP(1)= XGW7F1
      TMP(2)= YGW7F1
      TMP(3)= ZGW7F1
      CALL MULT31 (DCL1,TMP(1),TMP(4))
      UW77F = TMP(4)
      VW77F = TMP(5)
      WW77F = TMP(6)
1035  VA77F = SQRT((U777F-UW77F)**2+(V777F-VW77F)**2+(W777F-WW77F)**2)
      DYNPP = .5*RHOAS*VA77F**2
      AMACH = 0.
      IF (VS77F.NE.0.) AMACH=VA77F/VS77F
      TMP(1)= U777F-UW77F
      ALPHR = ATAN2(W777F-WW77F,TMP(1))
      BETAR = ATAN2(V777F-VW77F,TMP(1))
      ALPHD = ALPHR*57.2957795
      BETAD = BETAR*57.2957795
      IF (INDSKP.EQ.0) GO TO 1134
                       ALPL1 = ALPHR
                       BETL1 = BETAR
1134  IF (INDADD.EQ.0) GO TO 1163
      DELT1 = TIME-TL11
      IF (ABS(DELT1).LT.DLTM1) GO TO 1163
      ALPHR1= (ALPHR-ALPL1)/DELT1
      BETAR1= (BETAR-BETL1)/DELT1
                       ALPL1 = ALPHR
                       BETL1 = BETAR
                       TL11  = TIME
C1163 2SDF9
 1163 VG77F = SQRT(XG77F1**2+YG77F1**2+ZG77F1**2)
      IF (VG77F.EQ.0.) GO TO 1314
C1201 2SD10
 1201 TMP(1)=-ZG77F1/VG77F
      IF(ABS(TMP(1)).LT.1.)GO TO 1211
      GAM7R=1.5707963*TMP(1)/ABS(TMP(1))
      GO TO 1212
 1211 GAM7R=ASIN(TMP(1))
 1212 GAM7D = GAM7R*57.2957795
      GAMOR = GAM7R
```

```
      GAMDD = GAM7D
      SIG7R = ATAN2(YG77F1,XG77F1)
      SIGOR = SIG7R
      SIG7D = SIG7R*57.2957795
      SIGDD = SIG7D
C1236 25011
 1236 SGAMD = SIN(GAMOR)
      CGAMD = COS(GAMOR)
      SSIGD = SIN(SIGOR)
      CSIGD = COS(SIGOR)
      IF (INDFPR.EQ.0) GO TO 1314
      DELT2 = TIME-TL12
      IF (ABS(DELT2).LT.DLTM2)  GO TO 1314
      GAM7R1= (GAMOR-GAML1)/DELT2
      SIG7R1= (SIGOR-SIGL1)/DELT2
                            TL12 = TIME
                            GAML1= GAMOR
                            SIGL1= SIGOR
 1314 IF(INDWIN.NE.0)GO TO 1315
      SGAMA=SGAMD
      GO TO 1316
 1315 IF(VA77F.EQ.0.)GO TO 1316
      SGAMA=(ZG77F1-ZG77F1)/VA77F
 1316 IF(INDPLA.EQ.0)GO TO 1754
C1517 25015
 1517 IF (INDGRT.EQ.0) GO TO 1701
 1524 IF (DCL1.EQ.0.) GO TO 1536
      THTPR = ATAN2(-DCL3,DCL1)
      THTPD = THTPR*57.2957795
 1536 IF (ABS(DCL2).GT.1.) GO TO 1546
      PSIPR = ASIN(DCL2)
      GO TO 1554
 1546 PSIPR = 1.5707963*DCL2/ABS(DCL2)
 1554 PSIPD = PSIPR*57.2957795
      IF (DCM2.EQ.0.) GO TO 1701
      PHIPR = ATAN2(-DCN2,DCM2)
      PHIPD = PHIPR*57.2957795
 1701 IF (INDACM.EQ.0) GO TO 1754
      IF (AMASS.EQ.0.) CALL EXERR (14)
      AX77F = FXB7P/AMASS-GXB7F
      AY77F = FYB7P/AMASS-GYB7F
      AZ77F = FZB7P/AMASS-GZB7F
      IF (INDACM.NE.1) GO TO 1754
      TMP(1)= AX77F
      TMP(2)= AY77F
      TMP(3)= AZ77F
      CALL TRNPOS  (DCL1,TMP(19))
      CALL MULT31  (TMP(19),TMP(1),TMP(4))
      AXP7F = TMP(4)
      AYP7F = TMP(5)
      AZP7F = TMP(6)
 1754 CALL VPCS3
      CALL TFFS3
      WTP7P = AMASS*GREFF
C
C     CHUTE DRAG COMPUTATION
C       ICS=1 (FROM LANDING ROLL COMPUTATION,AUTS)
C       ICS=0 (INPUT)
C       CDCH DRAG COEFFICIENT
C       SCH REFERENCE AREA
C       XCH, YCH, ZCH = CHUTE ATTACH POINTS
C
 2061 IF(ICS.EQ.1)GO TO 2050
      FCX=0.
      FDC=0.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      FCY=0.
      FCZ=0.
      MDX=0.
      MDY=0.
      MDZ=0.
      GO TO 2062
 2050 FUC=CDCH*SSH*DYNPP
      FCX=-FUC*(U777F-UW77F)/VA77F
      FCY=-FUC*(V777F-VW77F)/VA77F
      FCZ=-FUC*(W777F-WW77F)/VA77F
      MDX=YCH*FCZ-ZCH*FCY
      MDY=ZCH*FCX-XCH*FCZ
      MDZ=XCH*FCY-YCH*FCX
 2062 CALL SACS3
 2125 CALL LGEAR3
      CALL FLEX3
      FXB7P  = TXB7P-AA77P  +AMASS*GXB7F+DLFXP+FXM+FCX
      FYB7P  = TYB7P+YA77P  +AMASS*GYB7F+DLFYP+FYM+FCY
      FZB7P  = TZB7P-ANA77P+AMASS*GZB7F+DLFZP+FZM+FCZ
      ALB77F= ALT77F+DLLTF+ALA77F+LM+MDX
      AMB77F= AMT77F+DLMTF+AMA77F+MM+MDY
      ANB77F= ANT77F+DLNTF+ANA77F+NM+MDZ
      IF (AMASS.EQ.0.) CALL EXERR (14)
      TMP(1)= 2.*AMASS1/AMASS
      U777F1= FXB7P/AMASS-QI77R*W777F+RI77R*V777F
      V777F1= FYB7P/AMASS-RI77R*U777F+PI77R*W777F+TMP(1)*RI77R*ALYJDF
      W777F1= FZB7F/AMASS-PI77R*V777F+QI77R*U777F-TMP(1)*QI77R*ALZJDF
      AKL1   = ALB77F+(AIYYBS-AIZZBS)*QI77R*RI77R
      AKM1   = AMB77F+(AIZZBS-AIXXBS)*PI77R*RI77R
      AKN1   = ANB77F+(AIXXBS-AIYYBS)*PI77R*QI77R
      IF (INDXZS.EQ.0) GO TO 2317
      AKL3   = 0.
      AKM3   = 0.
      AKN3   = 0.
      AKL4   = 0.
      AKM4   = 0.
      AKN4   = 0.
      GO TO 2370
C2317 2SD21
 2317 TMP(1)= QI77R**2
      TMP(2)= RI77R*PI77R
      AKL3   =-AIXYBS  *TMP(2)
      AKM3   = AIXYBS  *RI77R*QI77R
      AKN3   =-AIXYBS  *(TMP(1)-PI77R**2)
      AKL4   = AIYZBS  *(TMP(1)-RI77R**2)
      AKM4   =-AIYZBS  *PI77R*QI77R
      AKN4   = AIYZBS  *RI77R*PI77R
C2370 2SD22
 2370 IF (INDXYS.EQ.0) GO TO 2376
      AKL2   = 0.
      AKM2   = 0.
      AKN2   = 0.
      GO TO 2422
C2376 2SD23
 2376 AKL2   = AIXZBS*PI77R*QI77R
      AKM2   = AIXZBS*(RI77R**2-PI77R**2)
      AKN2   =-AIXZBS*QI77R*RI77R
C2422 2SD24
 2422 AKL5   =-AIXXS1*PI77R+AIXYS1*QI77R+AIXZS1*RI77R+AMASS1*ALLJDF**2*
     *        PI77R
      AKM5   =-AIYYS1*QI77R+AIXYS1*PI77R+AIYZS1*RI77R+AMASS1*ALMJDF**2*
     *        QI77R
      AKN5   =-AIZZS1*RI77R+AIYZS1*QI77R+AIXZS1*PI77R+AMASS1*ALNJDF**2*
     *        RI77R
C 16 2SD26
```

```
  316 IF (INDRMG.NE.0) GO TO 2524
      AKL5   = 0.
      AKM6   = 0.
      AKN6   = 0.
      GO TO 2615
C2524 25027
 2524 THTRR = THTRL*.01745329
      STHTR = SIN(THTRR)
      CTHTR = COS(THTRR)
      CALL TLU (TIMES,LOC(1),THTRR1)
      CALL INTEG (LTHTRR , THTRR1 )
      THTRD = THTRR*57.2957795
      OMGRR = OMGRT*.10471975
      AKL6   = AIXR7S*OMGRR*(QI77R+THTRR1)*STHTR
      AKM6   =-AIXR7S*OMGRR*(PI77R*STHTR+RI77R*CTHTR)
      AKN6   = AIXR7S*OMGRR*(QI77R+THTRR1)*CTHTR
C2615 25028
 2615 AKLT   = AKL1+AKL2+AKL3+AKL4+AKL5+AKL6
      AKMT   = AKM1+AKM2+AKM3+AKM4+AKM5+AKM6
      AKNT   = AKN1+AKN2+AKN3+AKN4+AKN5+AKN6
      TMP(1)= AIXXBS
      TMP(2)=-AIXYBS
      TMP(3)=-AIXZBS
      TMP(4)=-AIXYBS
      TMP(5)= AIYYBS
      TMP(6)=-AIYZBS
      TMP(7)=-AIXZBS
      TMP(8)=-AIYZBS
      TMP(9)= AIZZBS
      CALL INVR3 (TMP(1),TMP(10),INDEP)
      IF (INDEP.NE.1) CALL EXCPR (16)
      TMP(1)= AKLT
      TMP(2)= AKMT
      TMP(3)= AKNT
      CALL MULT31 (TMP(10),TMP(1),TMP(4))
      PI77R1= TMP(4)
      QI77R1= TMP(5)
      RI77R1= TMP(6)
      CALL INTEG (LA(10) , U777F1 )
      CALL INTEG (LA(11) , V777F1 )
      CALL INTEG (LA(12) , W777F1 )
      CALL INTEG (LA(13) , PI77R1 )
      CALL INTEG (LA(14) , QI77R1 )
      CALL INTEG (LA(15) , RI77R1 )
      PI77D = PI77R*57.2957795
      QI77D = QI77R*57.2957795
      RI77D = RI77R*57.2957795
      AL17S1= PI77R*DCM1-QI77R*DCN1
      AL27S1= RI77R*DCM2-QI77R*DCN2
      AL37S1= RI77R*DCM3-QI77R*DCN3
      AM17S1= PI77R*DCN1-RI77R*DCL1
      AM27S1= PI77R*DCN2-RI77R*DCL2
      AM37S1= PI77R*DCN3-RI77R*DCL3
      AN177S= QI77R*DCL1-PI77R*DCM1
      AN277S= QI77R*DCL2-PI77R*DCM2
      AN377S= QI77R*DCL3-PI77R*DCM3
      CALL INTEG (LA( 1) , AL17S1 )
      CALL INTEG (LA( 2) , AL27S1 )
      CALL INTEG (LA( 3) , AL37S1 )
      CALL INTEG (LA( 4) , AM17S1 )
      CALL INTEG (LA( 5) , AM27S1 )
      CALL INTEG (LA( 6) , AM37S1 )
      CALL INTEG (LA( 7) , AN177S )
      CALL INTEG (LA( 8) , AN277S )
      CALL INTEG (LA( 9) , AN377S )
```

```
      DCL1 = AL1
      DCL2 = AL2
      DCL3 = AL3
      DCM1 = AM1
      DCM2 = AM2
      DCM3 = AM3
      DCN1 = AN1
      DCN2 = AN2
      DCN3 = AN3
      CALL INVR3  (DCL1,TMP(1),INDER)
      IF (INDER.NE.1) CALL EXERR (17)
      CALL TRNPOS (TMP(1),TMP(10))
      AL1 = .5*(AL1+TMP(10))
      AL2 = .5*(AL2+TMP(13))
      AL3 = .5*(AL3+TMP(16))
      AM1 = .5*(AM1+TMP(11))
      AM2 = .5*(AM2+TMP(14))
      AM3 = .5*(AM3+TMF(17))
      AN1 = .5*(AN1+TMP(12))
      AN2 = .5*(AN2+TMP(15))
      AN3 = .5*(AN3+TMP(18))
      INDSKP=
      INDARC = HGC7F .GT. AMAXH
      CALL STGTST (INDSTG)
      IF (SW  .OR.  HGC7F.GT. 0.) RETURN
      INDSTE = 1
      WRITE (6,30)
   30 FORMAT (1H0,19X,10HSTOP-HGC7F)
      RETURN
C     **********
      ENTRY OPT5
C     **********
 3340 CALL STFL(3,1,H9CI)
      CALL STOVAR (8,TIME ,TIMES ,XG77F ,YG77F ,HGC7F,U777F,V777F,W777F)
      CALL STOVAR (8,PI77R,QI77R,RI77R,AMACH,VA77F ,DYNPP,XG77F1,YG77F1)
      CALL STFL(1,1,ZG77F1)
      IF(INDAPC.NE.0)CALL STOVAR(2,ALPHD,BETAD,DU,DU,DU,DU,DU,DU)
      IF(INDAD0.EQ.0)GO TO 3341
      ALPHD1=ALPHR1*57.2957795
      BETAD1=BETAR1*57.2957795
      CALL STOVAR(2,ALPHD1,BETAD1,DU,DU,DU,DU,DU,DU)
 3341 IF(INDGSP.NE.0)CALL STFL(1,1,VG77F)
      IF(INDFPA.NE.0)CALL STOVAR(2,GAM7D,SIG7D,DU,DU,DU,DU,DU,DU)
      IF(INDFPP.NE.0)CALL STOVAR(2,GAM7R1,SIG7R1,DU,DU,DU,DU,DU,DU)
      IF(INDGRT.NE.0)CALL STOVAR(3,THTPD,PSIPD,PHIPD,DU,DU,DU,DU,DU)
      IF(INDACM.EQ.1)CALL STOVAR(3,AXP7F,AYP7F,AZP7F,DU,DU,DU,DU,DU)
      IF(INDACM.EQ.2)CALL STOVAR(3,AX77F,AY77F,AZ77F,DU,DU,DU,DU,DU)
      IF(INDWGT.NE.0)CALL STFL(1,1,WTR7P)
      IF(INDRMC.NE.0)CALL STFL(1,1,THTRD)
      CALL STOVAR(4,FDC,FCX,FCY,FCZ,DU,DU,DU,DU)
      CALL LGEAR6
      CALL VPC56
      CALL TFF56
      CALL SACS6
      CALL FLEX6
      CALL STFL(0,1,DU)
      RETURN
C     **********
      ENTRY OPT7
C     **********
      TIME = DM(3)
      TIMES= DM(3) - TIMSX
      CALL UPDAT (5,LA(1 ) ,AL1,AL2,AL3,AM1,AM2 )
      CALL UPDAT (5,LA(6 ) ,AM3,AN1,AN2,AN3,U777F  )
      CALL UPDAT (5,LA(11) ,V777F,W777F,PI77R,QI77R,RI77R )
```

```
      CALL UPDAT(3,LA(16),XG77F,YG77F,ZG77F,DU,DU)
      IF(INORMC.NE.0) CALL UPDAT(1,LTATRR,TATRP,DU,DU,DU,DU)
      CALL LGEAR7
      CALL VPCS7
      CALL TFFS7
      CALL SAGS7
      CALL FLEX7
      RETURN
      END
      SUBROUTINE LGEAR1
      COMMON/DIRCOM/DM1(130),AMASS,DM2(147),DCL1,DCM1,
     CDCN1,DCL2,DCM2,DCN2,DCL3,DCM3,DCN3,DM3(99),FX87P,
     CDUM4(3),FY87P(4),FZ87P,DM5(17),GX87F,DM6(8),GZ87F,
     CDM7(218),INDSTE(235),
     CTIME,DM8(287),PI77R(2),PI77R1(2),DM9(4),
     CQI77P(2),QI77R1(2),DM10(4),RI77R(2),RI77R1(2),DM11(48),
     CXG77F(2),XG77F1(12),YG77F(2),YG77F1(12),
     CZG77F(2),ZG77F1(2),DUM13(52),
     CNSTRUT,MASS(5),RX(5),RY(5),RZ(5),THETAU(5),ERDEG,RGR,
     CNTIRES(5),FZERO(5),W(5),DELTAM(5),MOMENT(5),
     CRF(5),VZ   ,IFD,PZERO(5),VZERO(5),A(5),P2L(5),V2L(5),
     CA2(5),IL,S2T(5),ES2(5),C2L(5),MASS2(5),MUS(5),
     CCC(5),CE(5),C2C(5),C2L(5),NVGPT,NPP,MB(5),RLT,NDELTA,
     CSS(5),SB(5),SD21(2),SD22(2),SD23(2),SD24(2),SD25(2)
      COMMON/DIRCOM/
     CSD11(2),SD12(2),SD13(2),SD14(2),SD15(2),
     CS1(2),SS2(2),S3(2),S4(2),S5(2),
     CS2D21(2),S2D22(2),S2D23(2),S2D24(2),S2D25(2),
     CS2D11(2),S2D12(2),S2D13(2),S2D14(2),S2D15(2),
     CS21(2),S22(2),S23(2),S24(2),S25(2),
     COMTD11(2),OMTD12(2),OMTD13(2),OMTD14(2),OMTD15(2),
     COMT1(2),OMT2(2),OMT3(2),OMT4(2),OMT5(2),
     CAI(5),BI(5),DELTA1,DELTA2,DELTA3,DELTA4,DELTA5,
     CDDELT1,DDELT2,DDELT3,DDELT4,DDELT5,ISTAGE,
     CPRTMIN,IPLT,ISOF,ISTPL1,ISTPL2,ISTPL3,ISTPL4,ISTPL5,
     CDM14(22),IB(5),DM15(127),INDLG,DM16(107),CASK(44),
     *DM17,NMODE,DM18(40),SXMOD(100),SYMOD(100),SZMOD(100),DM19(1686),
     *GJDZ(20),DCM2.(20)
      REAL MASS,MOMENT,    MASS2,MUS,NTIRES,MB
      DIMENSION DLGAUT(14),DLGDE(47),DLG(7),DLGE(299)
      COMMON/LGAUTS/ARG11,ARG13,ARG31,ARG33,AMA(5),VAXLE(5)
      COMMON/LGDE/LA(25),FC2(5),P2(5),PRES(5),C(5),IPPT,LTFT
      COMMON/LG/FXM,FYM,FZM,LM,MM,NM,EPSLO2
      COMMON/LGE/A11(5),A13(5),A31(5),A33(5),RRGGX(5),
     CRL(3,3),RI(3,3,5),RAX(5),RAY(5),RAZ(5),TMP(3),ZZERO(5),
     CXR(5),YR(5),EPSLON(5),PA(5),FDELTA(5),
     CFTRZ(5),RUX(5),RUY(5),RUZ(5),RUXG(5),RUYG(5),RUZG(5),
     CVTX(5),VTY(5),VTZ(5),GZ(5),          VGPT(5),FTRX(5),FTRY(5),
     CDX(5),DY(5),DZ(5),FT(5),FDX(5),FDY(5),FF(5),AA(5),C2(5),
     CSR(5),SF(5),PSKD(5),MUVP(5),MTRX(5),MTRY(5),
     CMTRZ(5),MA(5),RG11,RG13,RG31,RG33,IPRT,
     CMTX,MTY,MTZ,SFTRX,SFTRY,SFTRZ,FTRA,
     CFTRA,FTRC,SMTRX,SMTRY,SMTRZ
      COMMON/FLXOP/GFORC2(100),GFORC3(100),GFORC4(100),BM1(308)
      EQUIVALENCE (DLGAUT(1),ARG11),(DLGDE(1),LA(1)),
     *(DLG(1),FXM),(DLGE(1),A11(1))
      COMMON/TABSPC/DUMM1(103),LOC(7)
      COMMON/HTCOM/HT,HT1,HT2
      REAL MUVP,MTRX,MTRY,MTRZ,MA,
     CMTX,MTY,MTZ,LM,MM,NM
      DIMENSION DELTA(5),DDELTA(5),P(5),
C           SD2(2,5),SD1(2,5),S(2,5),S2D2(2,5),S2D1(2,5),S2(2,5),
     COMETD1(2,5),OMET(2,5)
      EQUIVALENCE (P(1),PRES(1))
      EQUIVALENCE (DELTA(1),DELTA1),(DDELTA(1),DDELT1)
```

```
      EQUIVALENCE
     C(SD21(1),SD2(1,1)),(SD11(1),SD1(1,1)),(S1(1),S(1,1)),
     C(S2D21(1),S2D2(1,1)),(S2D11(1),S2D1(1,1)),(S21(1),S2(1,1)),
     C(OMTD11(1),OMETD1(1,1)),(OMT1(1),OMET(1,1))
C     PRE DATA INITIALIZATION
      DO 1 I=1,14
    1 DLGAUT(I)=0.
      DO 2 I=1,47
    2 DLGDE(I)=0.
      DO 3 I=1,7
    3 DLG(I)=0.
      DO 4 I=1,299
    4 DLGE(I)=0.
      ISTAGE=0
      IPLT=0
      IPPT=0
      LTPT=1
      ISDF=0
      ISTPL1=0
      ISTPL2=0
      ISTPL3=0
      ISTPL4=0
      ISTPL5=0
      EPSL02=0.
      RETURN
C     **********
      ENTRY LGEAR2
C     **********
      FXM=0.
      FYM=0.
      FZM=0.
      LM=0.
      MM=0.
      NM=0.
      ERR=.01745329*ERDEG
      RG11=COS(ERR)
      RG13=-SIN(ERR)
      RG31=-RG13
      RG33=RG11
      ARG11=RG11
      ARG13=RG13
      ARG31=RG31
      ARG33=RG33
      IF(INDLG.EQ.0)RETURN
C     INITIALIZATION AFTER DATA READ IN
C       VARIABLES TO BE INTEGRATED
C          (OMTD11,OMT1   )
C          (OMTD12,OMT2   )
C          (OMTD13,OMT3   )
C          (OMTD14,OMT4   )
C          (OMTD15,OMT5   )
C          (SD21   ,SD11   )          (SD11   ,S1     )
C          (SD22   ,SD12   )          (SD12   ,S2     )
C          (SD23   ,SD13   )          (SD13   ,S3     )
C          (SD24   ,SD14   )          (SD14   ,S4     )
C          (SD25   ,SD15   )          (SD15   ,S5     )
C          (S2D21  ,S2D11  )          (S2D11  ,S21    )
C          (S2D22  ,S2D12  )          (S2D12  ,S22    )
C          (S2D23  ,S2D13  )          (S2D13  ,S23    )
C          (S2D24  ,S2D14  )          (S2D14  ,S24    )
C          (S2D25  ,S2D15  )          (S2D15  ,S25    )
      IF(INDLG.EQ.2)GO TO 5
      NDEQ=5*NSTRUT
      IF(IL.NE.0)NDEQ=3*NSTRUT
      CALL INUPJ(NDEQ,LA)
```

```
      DO 10 I=1,NSTRUT
      THETAR=.01745329*THETAD(I)
      A11(I)=COS(THETAR)
      A13(I)=-SIN(THETAR)
      A31(I)=-A13(I)
      A33(I)=A11(I)
 10   RRCGX(I)=A11(I)*RX(I)+A13(I)*RZ(I)
      IPRT=0
      RETURN
C     ***********
C     INITIAL PRINT
      ENTRY LGEAR4
C     ***********
      RETURN
C     ***********
      ENTRY LGEAR3
C     ***********
C     MAIN COMPUTATIONAL AREA
C     RL MATRIX ELEMENTS
      IF(INULG.EQ.1)RETURN
      RL(1,1)=DCL1*RG11+DCL3*RG13
      RL(1,2)=DCL2
      RL(1,3)=DCL1*RG31+DCL3*RG33
      RL(2,1)=DCM1*RG11+DCM3*RG13
      RL(2,2)=DCM2
      RL(2,3)=DCM1*RG31+DCM3*RG33
      RL(3,1)=DCN1*RG11+DCN3*RG13
      RL(3,2)=DCN2
      RL(3,3)=DCN1*RG31+DCN3*RG33
      CALL LGEAR5
C     CALCULATION OF S2D2, S2D1, AND S2
      DO 100 I=1,NSTRUT
      IF(INULG.EQ.2)GO TO 20
      IF(IL.NE.0)GO TO 59
      TMP(1)=SD1(1,I)-A2(I)*S2D1(1,I)/A(I)
      S2D2(1,I)=((P(I)-P2(I))*A2(I)+C(I)*TMP(1)*ABS(TMP(1))
     C-S2D1(1,I)*(C2(I)*ABS(S2D1(1,I))+C2L(I)))
     C/MASS2(I)+G2(I)-SR(I)+SD2(1,I)
      HT2=HT
      IF(S2D1(1,I))86,87,88
 86   TTIME=S2(1,I)/ABS(S2D1(1,I))
 89   IF(TTIME.GE.HT)GO TO 87
      HT2=TTIME
      GO TO 87
 88   TTIME=(S2T(I)-S2(1,I))/S2D1(1,I)
      GO TO 89
 87   CONTINUE
      IF(S2(1,I).GT.(-ES2(I)))GO TO 58
      WRITE(6,57)I,I,S2(1,I)
 57   FORMAT(58X,5H-ES2(,I1,10H) EXCEEDED/
     C58X,3HS2(,I1,4H) = E15.7)
 58   IF(S2(1,I).LE.ES2(I))GO TO 61
      IF(S2(1,I).LE.(S2T(I)-ES2(I)))GO TO 141
      IF(S2(1,I).LE.(S2T(I)+ES2(I)))GO TO 62
      WRITE(6,63)I,I,S2(1,I)
 63   FORMAT(58X,5H ES2(,I1,10H) EXCEEDED/
     C58X,3HS2(,I1,4H) = E15.7)
 62   IF(S2D1(1,I).GT.0.)S2D1(1,I)=0.
      IF(S2D2(1,I).LT.0.)GO TO 141
      GO TO 140
 61   IF(S2D1(1,I).LT.0.)S2D1(1,I)=0.
      IF(S2D2(1,I).LT.0.)GO TO 140
 141  FC2(I)=0.
      GO TO 60
 140  FC2(I)=-MASS2(I)*S2D2(1,I)
```

```
      S2D2(1,I)=0.
      GO TO 60
 59   S2D2(1,I)=0.
      S2D1(1,I)=0.
      S2(1,I)=0.
      FC2(I)=-(P(I)-P2(I))*A2(I)-C(I)*SD1(1,I)*ABS(SD1(1,I))
      GO TO 94
 60   CONTINUE
      I1=2*I+3*NSTRUT-1
      I2=I1+1
      CALL INTEG(LA(I1),S2D2(1,I))
      CALL INTEG(LA(I2),S2D1(1,I))
 94   CONTINUE
C     COMPUTATION OF AA(I)
      TMP(2)=0.
      IF(SD1(1,I).NE.0.)TMP(2)=SD1(1,I)/ABS(SD1(1,I))
      SF(I)=-P(I)*(A(I)-A2(I))-P2(I)*A2(I)+FC2(I)-S2D1(1,I)*
     C(C2(I)*ABS(S2D1(1,I))+C2L(I))-FF(I)*TMP(2)
      AA(I)=(FT(I)+SF(I))/MASS(I)
C     CALCULATION OF SD2, SD1, AND S
C     CALCULATION OF OMETD1(I) AND OMET(I)
      SD2(1,I)=SR(I)+AA(I)-GZ(I)
      HT1=HT
      IF(SD1(1,I))76,77,78
 76   TTIME=S(1,I)/ABS(SD1(1,I))
 79   IF(TTIME.GE.HT)GO TO 77
      HT1=TTIME
      GO TO 77
 78   TTIME=(S0(I)-S(1,I))/SD1(1,I)
      GO TO 79
 77   CONTINUE
      IF(S(1,I).GT.(-ES(I)))GO TO 50
      WRITE(6,49)I,I,S(1,I)
 49   FORMAT(58X,4H-ES(,I1,10H) EXCEEDED/
     C58X,2HS(,I1,4H) = E15.7)
 50   IF(S(1,I).LE.ES(I))GO TO 51
      IF(S(1,I).LE.(S0(I)-ES(I)))GO TO 55
      IF(S(1,I).LE.(S0(I)+ES(I)))GO TO 52
      WRITE(6,53)I,I,S(1,I)
 53   FORMAT(58X,4H ES(,I1,10H) EXCEEDED/
     C58X,2HS(,I1,4H) = E15.7)
 52   IF(SD1(1,I).GT.0.)SD1(1,I)=0.
      IF(SD2(1,I).LT.0.)GO TO 55
      SD2(1,I)=0.
      GO TO 55
 51   IF(SD1(1,I).LT.0.)SD1(1,I)=0.
      IF(SD2(1,I).LT.0.)SD2(1,I)=0.
 55   CONTINUE
      I2=2*I+NSTRUT-1
      I1=I2+1
      CALL INTEG(LA(I2),SD2(1,I))
      CALL INTEG(LA(I1),SD1(1,I))
 20   TMP(1)=RZERO(I)-DELTA(I)
      IF(CASK(I).GT.1.E-10)GO TO 200
      MA(I)=-FTRY(I)*TMP(1)*RI(2,1,I)+FTRX(I)*TMP(1)
     C*RI(2,2,I)
      GO TO 201
 200  MA(I)=TMP(1)*SQRT(FTRY(I)*FTRY(I)+FTRX(I)*FTRX(I))
      MA(I)=SIGN(MA(I),-VAXLE(I)-OMET(1,I)*TMP(1))
 201  AMA(I)=MA(I)
      IF(INDLG.EQ.2)GO TO 100
      IF(IB(I).NE.(-1))GO TO 48
      OMETD1(1,I)=0.
      OMET(1,I)=0.
      GO TO 21
```

```
    9  TMP(1)=0.
       IF(OMET(1,I).NE.0.)TMP(1)=OMET(1,I)/ABS(OMET(1,I))
       OMETD1(1,I)=(MA(I)-MB(I)*TMP(1))/(NTIRES(I)*MOMENT(I))
   21  CALL INTEG(LA(I),OMETD1(1,I))
  100  CONTINUE
C      CALCULATION OF FTRA, FTRB, AND FTRC
       SFTRX=0.
       SFTRY=0.
       SFTRZ=0.
       DO 70 I=1,NSTRUT
       SFTRX=SFTRX+FTRX(I)
       SFTRY=SFTRY+FTRY(I)
   70  SFTRZ=SFTRZ+FTRZ(I)
       FTRA=RL(1,1)*SFTRX+RL(1,2)*SFTRY+RL(1,3)*SFTRZ
       FTRB=RL(2,1)*SFTRX+RL(2,2)*SFTRY+RL(2,3)*SFTRZ
       FTRC=RL(3,1)*SFTRX+RL(3,2)*SFTRY+RL(3,3)*SFTRZ
C      CALCULATION OF MTX, MTY, AND MTZ
       SMTRX=0.
       SMTRY=0.
       SMTRZ=0.
       DO 75 I=1,NSTRUT
       SMTRX=SMTRX+MTRX(I)
       SMTRY=SMTRY+MTRY(I)
   75  SMTRZ=SMTRZ+MTRZ(I)
       MTX=RL(1,1)*SMTRX+RL(1,2)*SMTRY+RL(1,3)*SMTRZ
       MTY=RL(2,1)*SMTRX+RL(2,2)*SMTRY+RL(2,3)*SMTRZ
       MTZ=RL(3,1)*SMTRX+RL(3,2)*SMTRY+RL(3,3)*SMTRZ
C      CALCULATION OF FXM, FYM, FZM, LM, MM, AND NM
       FYM=0.
       FZM=0.
       FXM=0.
       LM=0.
       MM=0.
       NM=0.
       BFX=0.
       BFY=0.
       BFZ=0.
       BLM=0.
       BMM=0.
       BNM=0.
       DO 82 I=1,NSTRUT
       DFXM=0.
       DFYM=0.
       DFZM=0.
       DLM=0.
       DMM=0.
       DNM=0.
       DO 14 IL4=1,NMODE
       NBB=(IL4-1)*NSTRUT+I
       DFXM=DFXM-MASS(I)*SXMOD(NBB)*GQD2(IL4)
       DFYM=DFYM-MASS(I)*SYMOD(NBB)*GQD2(IL4)
       DFZM=DFZM-MASS(I)*SZMOD(NBB)*GQD2(IL4)
       NBH=(I-1)*NMODE+IL4
       DLM=DLM-MASS(I)*GFORC2(NBH)*GQD2(IL4)
       DMM=DMM-MASS(I)*GFORC3(NBH)*GQD2(IL4)
   14  DNM=DNM-MASS(I)*GFORC4(NBH)*GQD2(IL4)
       BFX=BFX+DFXM
       BFY=BFY+DFYM
       BFZ=BFZ+DFZM
       BLM=BLM+DLM
       BMM=BMM+DMM
       BNM=BNM+DNM
       TMP(1)=MASS(I)*SD2(1,I)
       FXM=FXM+TMP(1)*A31(I)
       FYM=FYM
```

```
      FZM=FZM+TMP(1)*A33(I)
      LM=LM+TMP(1)*A11(I)*RY(I)
      MM=MM-TMP(1)*RRCGX(I)
82    NM=NM+TMP(1)*A13(I)*RY(I)
      FXM=FXM+BFX+FTRA
      FYM=FYM+BFY+FTRB
      FZM=FZM+BFZ+FTRC
      LM=LM+BLM+MTX
      MM=MM+BMM+MTY
      NM=NM+BNM+MTZ
      RETURN
      END
      SUBROUTINE LGEA3C
      COMMON/DIRCOM/DM1(136),AMASS,DM2(147),DCL1,DCM1,
     CDCN1,DCL2,DCM2,DCN2,DCL3,DCM3,DCN3,DM3(99),FX07P,
     CDUM4(3),FY07P(4),FZ07P,DM5(17),GX07F,DM6(8),GZ07F,
     CDM7(186),INDC.1,INDC.2,INDC.3,INDC.4,DM15(28),INDSTE,DM18(234),
     CTIME,DM8(287),PI77K,DM16,PI77R1,DM9(5),
     CQI77R,DM24,QI77R1,DM10(5),RI77R,DM17,RI77R1,DM11(49),
     CXG77F,DM19,XG77F1,DM20(11),YG77F,DM21,YG77F1,DM22(11),
     CZG77F,DM23,ZG77F1,DUM13(46),NSMAIN,NS2NDY(6),
     CNSTRUT,MASS(5),RX(5),RY(5),RZ(5),THETAD(5),ERDEG,RGR,
     CNTIRES(5),RZERO(5),W(5),DELTAM(5),MOMENT(5),
     CRF(5),VZ   ,IFD,PZERO(5),VZERO(5),A(5),P20(5),V20(5),
     CA2(5),IL,S2T(5),ES2(5),C2L(5),MASS2(5),MUS(5),
     CCC(5),CE(5),C2C(5),C2E(5),NVGPT,NPP,MB(5),RLT,NDELTA,
     CES(5),SD(5),SD21(2),SD22(2),SD23(2),SD24(2),SD25(2)
      COMMON/DIRCOM/
     CSD11(2),SD12(2),SD13(2),SD14(2),SD15(2),
     CS1(2),S2(2),S3(2),S4(2),S5(2),
     CS2D21(2),S2D22(2),S2D23(2),S2D24(2),S2D25(2),
     CS2D11(2),S2D12(2),S2D13(2),S2D14(2),S2D15(2),
     CS21(2),S22(2),S23(2),S24(2),S25(2),
     COMTD11(2),OMTD12(2),OMTD13(2),OMTD14(2),OMTD15(2),
     COMT1(2),OMT2(2),OMT3(2),OMT4(2),OMT5(2),
     CAI(5),BI(5),DELTA1,DELTA2,DELTA3,DELTA4,DELTA5,
     CDDELT1,DDELT2,DDELT3,DDELT4,DDELT5,ISTAGE,
     CPRTMIN,IPLT,ISDF,ISTPL1,ISTPL2,ISTPL3,ISTPL4,ISTPL5,
     COM14(262),CASK(5),DM25(39),DM26,NMODE,DM27(40),SXMOD(100),
     *SYMOD(100),SZMOD(100),DM28(1648),GQ(20),GQD1(20),GQD2(20),
     *DDM29(20)
      REAL MASS,MOMENT,   MASS2,MUS,NTIRES,MB
      COMMON/LGAUTS/ARG11,ARG13,ARG31,ARG33,AMA(5),VAXLE(5)
      COMMON/LGDE/LA(25),FC2(5),P2(5),PRES(5),C(5),IPPT,LTPT
      COMMON/LG/FXM,FYM,FZM,LM,MM,NM,EPSLO2
      COMMON/LGE/A11(5),A13(5),A31(5),A33(5),RRCGX(5),
     CRL(3,3),RI(3,3,5),RAX(5),RAY(5),RAZ(5),TMP(3),ZZERO(5),
     CXR(5),YR(5),EPSLON(5),PA(5),FDELTA(5),
     CFTRZ(5),RDX(5),RDY(5),RDZ(5),RDXG(5),RDYG(5),RDZG(5),
     CVTX(5),VTY(5),VTZ(5),GZ(5),          VGPT(5),FTRX(5),FTRY(5),
     CDX(5),DY(5),DZ(5),FT(5),FDX(5),FDY(5),FF(5),AA(5),C2(5),
     CSR(5),SF(5),PSKD(5),MUVR(5),MTRX(5),MTRY(5),
     CMTRZ(5),MA(5),RG11,RG13,RG31,RG33, IPRT,
     CMTX,MTY,MTZ,SFTRX,SFTRY,SFTRZ,FTRA,
     CFTRB,FTRC,SMTRX,SMTRY,SMTRZ
      COMMON/TABSPC/DUMM1(123),LOC(7)
      REAL MUVP,MTRX,MTRY,MTRZ,MA,
     CMTX,MTY,MTZ,LM,MM,NM
      DIMENSION DELTA(5),DDELTA(5),P(5),
C          SD2(2,5),SD1(2,5),S(2,5),S2D2(2,5),S2D1(2,5),S2(2,5),
     COMETD1(2,5),OMET(2,5)
      DIMENSION TEMP1(3),TEMP2(3)
      EQUIVALENCE  (P(1),PRES(1))
      EQUIVALENCE  (DELTA(1),DELTA1),(DDELTA(1),DDELT1)
      EQUIVALENCE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     C(SD21(1),SD2(1,1)),(SD11(1),SD1(1,1)),(S1(1),S(1,1)),
     C(S2D21(1),S2D2(1,1)),(S2D11(1),S2D1(1,1)),(S21(1),S2(1,1)),
     C(OMTD1(1),OMETD(1,1)),(OMT1(1),OMET(1,1))
      DO 13 I=1,NSTRUT
      FI=I
C     RI MATRIX ELEMENTS
      DO 20 J=1,3
      TEMP1(J)=0.
      TEMP2(J)=0.
      RI(1,J,I)=A11(I)*RL(1,J)+A13(I)*RL(3,J)
      RI(2,J,I)=RL(2,J)
   20 RI(3,J,I)=A31(I)*RL(1,J)+A33(I)*RL(3,J)
C     POSITION DEPENDENT CALCULATIONS
      DO 11 ILP=1,NMODE
      NOD=(ILP-1)*NSTRUT+I
      TEMP1(1)=TEMP1(1)+SXMOD(NOD)*GQ(ILP)
      TEMP1(2)=TEMP1(2)+SYMOD(NOD)*GQ(ILP)
   11 TEMP1(3)=TEMP1(3)+SZMOD(NOD)*GQ(ILP)
      TMP(1)=RF(I)-S(1,I)
      TMP(2)=RX(I)+A31(I)*TMP(1)+TEMP1(1)
      TMP(3)=RZ(I)+A33(I)*TMP(1)+TEMP1(3)
      RAX(I)=DCL1*TMP(2)+DCM1*(RY(I)+TEMP1(2))+DCN1*TMP(3)
      RAY(I)=DCL2*TMP(2)+DCM2*(RY(I)+TEMP1(2))+DCN2*TMP(3)
      RAZ(I)=DCL3*TMP(2)+DCM3*(RY(I)+TEMP1(2))+DCN3*TMP(3)
      TMP(1)=(XG77F-RGP+RAX(I))
      TMP(2)=ZG77F+RAZ(I)
      ZZERO(I)=TMP(1)*RG31+TMP(2)*RG33
      XR(I)=TMP(1)*RG11+TMP(2)*RG13
      YR(I)=YG77F  +RAY(I)
      IF(XR(I).LE.0.)GO TO 25
      IF(XR(I).GE.RLT)GO TO 25
      CALL TLU(XR(I),LOC(2),EPSLON(I))
      GO TO 24
   25 EPSLON(I)=0.
   24 DELTA(I)=RZERO(I)+ZZERO(I)+EPSLON(I)
      DDELTA(I)=DELTA(I)
      IF(DELTA(I).LE.0.)GO TO 26
      IF(DELTA(I).LE.DELTAM(I))GO TO 27
      WRITE(6,28)I,DELTA(I)
   28 FORMAT(36X,24HTIRE DEFLECTION EXCEEDED,10X,6HDELTA(,I1,4H) = ,
     CE15.7)
      CALL LOEASP
      INDSTE=0
      RETURN
   26 DELTA(I)=0.
   27 CONTINUE
      P(I)=PZERO(I)*VZERO(I)/(VZERO(I)+A2(I)*S2(1,I)-
     CS(1,I)*A(I))
      P2(I)=0.
      TMP(1)=V20(I)-A2(I)*S2(1,I)
      IF(TMP(1).EQ.0.)GO TO 31
      P2(I)=P20(I)*V20(I)/TMP(1)
   31 IF(IFD.EQ.0)GO TO 32
      FDELTA(I)=0.
      IF(DELTA(I).NE.0.)FDELTA(I)=AI(I)*(DELTA(I))**BI(I)
      GO TO 33
   32 CALL HIHC(3,LOC(1),NDELTA,NSTRUT,DU,DU,DELTA(I),FI,DU,DU,FDELTA(I)
     1)
   33 FTRZ(I)=-NTIRES(I)*FDELTA(I)
C     CALCULATION OF COMPONENTS OF GROUND PLANE
C     VELOCITIES VTX(I) AND VTY(I)
      TMP(1)=(RF(I)-S(1,I))
      DO 12 IL2=1,NMODE
      NO=(IL2-1)*NSTRUT+I
      TEMP2(1)=TEMP2(1)+(QI77R*SZMOD(NO)-RI77R*SYMOD(NO))*GQ(IL2)
```

```
     C+SXMOD(ND)*GQQ1(IL2)
      TEMP2(2)=TEMP2(2)+(RI77R*SXMOD(ND)-PI77R*SZMOD(ND))*GQ(IL2)
     C+SYMOD(ND)*GQQ1(IL2)
12    TEMP2(3)=TEMP2(3)+(PI77R*SYMOD(ND)-QI77R*SXMOD(ND))*GQ(IL2)
     C+SZMOD(ND)*GQQ1(IL2)
      RDX(I)=-SD1(1,I)*A31(I)+QI77R*(TMP(1)*A33(I)+RZ(I))-RI77R
     C*RY(I)+TEMP2(1)
      RDY(I)=TMP(1)*(RI77R*A31(I)-PI77R*A33(I))+RI77R*RX(I)-PI77R
     C*RZ(I)+TEMP2(2)
      RDZ(I)=-SD1(1,I)*A33(I)-TMP(1)*QI77R*A31(I)+PI77R*RY(I)-QI77R
     C*RX(I)+TEMP2(3)
      RDXG(I)=XG77F1+DCL1*RDX(I)+DCM1*RDY(I)+DCN1*RDZ(I)
      RDYG(I)=YG77F1+DCL2*RDX(I)+DCM2*RDY(I)+DCN2*RDZ(I)
      RDZG(I)=ZG77F1+DCL3*RDX(I)+DCM3*RDY(I)+DCN3*RDZ(I)
      TMP(1)=RZERO(I)-DELTA(I)
      TMP(2)=OMET(1,I)*RI(2,2,I)*TMP(1)
      TMP(3)=OMET(2,I)*RI(2,1,I)*TMP(1)
      VTX(I)=RG11*RDXG(I)+RG13*RDZG(I)+TMP(2)
      VTY(I)=RDYG(I)-TMP(3)
      VTZ(I)=RG31*RDXG(I)+RG33*RDZG(I)
C     CALCULATION OF GROUND PLANE FORCES FTRX(I) AND FTRY(I)
      TMP(1)=RG11*RDXG(I)+RG13*RDZG(I)
      VAXLE(I)=SQRT(TMP(1)*TMP(1)+RDYG(I)*RDYG(I))
      IF(CASK(I).GT.1.E-10)GO TO 34
      VGPT(I)=SQRT(VTX(I)*VTX(I)+VTY(I)*VTY(I))
      GO TO 35
34    TMP(1)=VAXLE(I)+OMET(1,I)*(RZERO(I)-DELTA(I))
      VGPT(I)=ABS(TMP(1))
35    IF(VGPT(I).LE.VZ)GO TO 40
      PSKD(I)=0.
      IF(ABS(VAXLE(I)).LE.1.E-8)GO TO 42
      PSKD(I)=VGPT(I)/VAXLE(I)
42    CALL TLU(PSKD(I),LOC(3),MUVP(I))
      FTRX(I)=MUVP(I)*FTRZ(I)/VGPT(I)
      IF(CASK(I).GT.1.E-10)GO TO 50
      TMP(3)=VTY(I)
      GO TO 51
50    TMP(3)=(VTY(I)+TMP(3))/VAXLE(I)*TMP(1)
51    FTRY(I)=FTRX(I)*TMP(3)
      IF(CASK(I).GT.1.E-10)GO TO 52
      TMP(2)=VTX(I)
      GO TO 53
52    TMP(2)=(VTX(I)-TMP(2))/VAXLE(I)*TMP(1)
53    FTRX(I)=FTRX(I)*TMP(2)
      GO TO 41
40    FTRX(I)=0.
      FTRY(I)=0.
41    CONTINUE
C     CALCULATION OF MTRX(I), MTRY(I), AND MTRZ(I)
      DX(I)=RG11*RAX(I)+RG13*RAZ(I)
      DY(I)=RAY(I)
      DZ(I)=RG31*RAX(I)+RG33*RAZ(I)+RZERO(I)-DELTA(I)
      MTRX(I)=DY(I)*FTRZ(I)-DZ(I)*FTRY(I)
      MTRY(I)=DZ(I)*FTRX(I)-DX(I)*FTRZ(I)
      MTRZ(I)=DX(I)*FTRY(I)-DY(I)*FTRX(I)
C     CALCULATION OF FT(I)
      FT(I)=-FTRX(I)*RI(3,1,I)-FTRY(I)*RI(3,2,I)
     C-FTRZ(I)*RI(3,3,I)
C     CALCULATION OF FF(I)
      FDX(I)=RI(1,1,I)*FTRX(I)+RI(1,2,I)*FTRY(I)+RI(1,3,I)*FTRZ(I)
      FDY(I)=RI(2,1,I)*FTRX(I)+RI(2,2,I)*FTRY(I)+RI(2,3,I)*FTRZ(I)
      FF(I)=MUS(I)*SQRT(FDX(I)*FDX(I)+FDY(I)*FDY(I))
C     CALULATION OF GZ(I)
      TMP(3)=-AZ(I)/A(I)
      C(I)=0.
```

```
      C2(I)=0.
      IF((SD1(1,I)+TMP(3)*S2D1(1,I)).GE.0.)GO TO 43
      IF(INDC02.EQ.0)GO TO 44
      CALL HIMO(3,LOC(5),NSMAIN,NSTRUT,DU,DU,S(1,I),FI,DU,DU,C(I))
      GO TO 44
 43   IF(INDC01.EQ.0)GO TO 44
      CALL HIMO(3,LOC(4),NSMAIN,NSTRUT,DU,DU,S(1,I),FI,DU,DU,C(I))
 44   IF(S2D1(1,I).GE.0.)GO TO 45
      IF(INDC04.EQ.0)GO TO 46
      CALL HIMO(3,LOC(7),NS2NDY,NSTRUT,DU,DU,S(1,I),FI,DU,DU,C2(I))
      GO TO 46
 45   IF(INDC03.EQ.0)GO TO 46
      CALL HIMO(3,LOC(6),NS2NDY,NSTRUT,DU,DU,S(1,I),FI,DU,DU,C2(I))
 46   GZ(I)=A31(I)*GXB7F+A33(I)*GZ07F
C     COMPUTATION OF SR(I)
      TEMP3=0.
      TMP(1)=RX(I)*RI77R-RZ(I)*PI77R
      DO 13 IL3=1,NMODE
      NDO=(IL3-1)*NSTRUT+I
 13   TEMP3=TEMP3+A31(I)*SXMOD(NDO)*GOD2(IL3)+A33(I)*SZMOD(NDO)
     C*GOD2(IL3)
      SR(I)    =A31(I)*FX07P/AMASS+A33(I)*FZ07P/AMASS
     C+A31(I)*(RZ(I)*QI77R1-RY(I)*RI77R1)+A33(I)
     C*(RY(I)*PI77R1-RX(I)*QI77R1)+A31(I)*(QI77R*
     C(RY(I)*PI77R-RX(I)*QI77R)-RI77R*TMP(1))
     C+A33(I)*(PI77R*TMP(1)-QI77R*(RZ(I)*QI77R
     C-RY(I)*RI77R))+TEMP3
 100  CONTINUE
C         SAVE EPSLON(2) FOR SACS SUBROUTINE
      EPSLO2=EPSLON(2)
      RETURN
      END
      SUBROUTINE FLEX1
C
C    SUBROUTINE FLEX1 EMPLOYS A MATRIX APPROACH TO DETERMINE THE DYNAMIC
C           RESPONSE OF SELECTED POINTS ON A FLEXIBLE AIRPLANE
C
      COMMON/DIRCOM/SKIPUP,DUD(7),AMINCR,DM1(13),AA77P,DM2(81),ALA77F,
     *DM3(35),AMA77F,DM4(9),ANA77F,ANA77P,DM5(12),AX77F,DM6,AY77F,
     *DM7(2),AZ77F,DM8(8..),YA77P,DM9(183),PI77R,DM10,PI77R1,DM11(5),
     *QI77P,DM12,QI77R1,DM13(5),RI77R,DM14,RI77R1,DM15(15),
     *U777F,DM16(19),V777F,DM17(3),W777F,DM18(93),NSTRUT,
     *MASS(5),RX(5),RY(5),RZ(5),DM19(32),RF(5),DM20(97),SD21(2),
     *SD22(2),SD23(2),SD24(2),SD25(2),DM21(99),IN,DM22(15.),
     *INDLG,DM23(151),INDFLX,NMODE,GMASS1(20),GFREQ(20),
     *SXMOD(100),SYMOD(100),SZMOD(100),SKC(5),TXMOD(50),
     *DCMODE(50),ARMCDE(120),NPTS,OUTMOD(120),ROIS(60),
     *PF(120),GD(20),GOD1(20),GOD2(20),DUM24(20)
      COMMON/FLXOP/GFORC2(100),GFORC3(100),GFORC4(100),
     *T(5),FCX,FCY,FCZ,XD2F(20),XD2T(20),YD2F(20),YD2T(20),ZD2F(20),
     *ZD2T(20),XD1F(20),XD1T(20),YD1F(20),YD1T(20),ZD1F(20),ZD1T(20),
     *XD0F(20),YD0F(20),ZD0F(20)
      COMMON/LGC/A11(5),A13(5),A31(5),A33(5),DM24(279)
      COMMON/HTCOM/HT,DOMM(2)
      REAL MASS
      DIMENSION GMASS(400),SMASS(25),GSMOD(400),COPMAS(20,20),
     *QS(20),QS1(20),GFORC1(60),DIFF(5),RZA11(25),RXA13(25),RKSY(25),
     *SDD(10),VARY1(3),OMXD1M(5),OMYD1M(5),OMZD1M(5),GF(20),FDC(3),
     *CTMP1(120),GTF(20),SD0(3),SD1(3),SD2(3),TITL(16),LA(40),
     *COEF(20,3)
      EQUIVALENCE (SD21(1),SDD(1))
      DATA HDR/5H  FLEX/,(TITL(I),I=1,16)/5HPOINT,4HXD2F,4HXD2T,4HYD2F,
     *4HYD2T,4HZD2F,4HZD2T,4HXD1F,4HXD1T,4HYD1F,4HYD1T,4HZD1F,
     *4HZD1T,4HXD0F,4HYD0F,4HZD0F/
C   PRE DATA INITIALIZATION
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 1 I=1,400
    1 GMASS(I)=0.
      DO 2 I=1,25
      SMASS(I)=0.
      RZA11(I)=0.
      RXA13(I)=0.
    2 RKSY(I)=0.
      DO 15 I=1,40
   15 LA(I)=0
      NMODE=1
      RETURN
C
      ENTRY FLEX2
C
      IF(INDFLX.EQ.0.OR.INDLG.EQ.0)RETURN
      HT=AMINER
      NVAR=2*NMODE
      CALL INUPD(NVAR,LA)
C
C  FORMULATE TIME INVARIANT ARRAYS
C
C  FORM COUPLED GENERALIZED MASS MATRIX
C
      DO 30 K=1,NMODE
      NA=(K-1)*NMODE+K
   30 GMASS(NA)=GMASS1(K)
      DO 31 N=1,NSTRUT
      LLA=(N-1)*NSTRUT+N
   31 SMASS(LLA)=MASS(N)
      CALL GTPRD(SXMOD,SMASS,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GMPRD(CTMP1,SXMOD,GSMOD,NMODE,NSTRUT,NMODE)
      CALL GMADD(GSMOD,GMASS,GMASS,NMODE,NMODE)
      CALL GTPRD(SYMOD,SMASS,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GMPRD(CTMP1,SYMOD,GSMOD,NMODE,NSTRUT,NMODE)
      CALL GMADD(GSMOD,GMASS,GMASS,NMODE,NMODE)
      CALL GTPRD(SZMOD,SMASS,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GMPRD(CTMP1,SZMOD,GSMOD,NMODE,NSTRUT,NMODE)
      CALL GMADD(GSMOD,GMASS,GMASS,NMODE,NMODE)
      CALL ARAY(1,NMODE,NMODE,20,20,GMASS,COPMAS)
      CALL DECOMP(NMODE,COPMAS,COEF)
C
C  FORM CONSTANT COMPONETS OF TANGENTIAL ACCELERATION
C
      CALL GTPRD(SXMOD,MASS,QS,NSTRUT,NMODE,1)
      CALL GTPRD(SYMOD,MASS,QS1,NSTRUT,NMODE,1)
      CALL CTIE(QS,QS1,CTMP1,NMODE,1,0,0,1)
      CALL GTPRD(SZMOD,MASS,QS,NSTRUT,NMODE,1)
      CALL CTIE(CTMP1,QS,GFORC1,NMODE,2,0,0,1)
      DO 33 L=1,NSTRUT
      DIFF(L)=RF(L)-SKC(L)
      MA=(L-1)*NSTRUT+L
      RZA11(MA)=RZ(L)+DIFF(L)*A11(L)
      RXA13(MA)=RX(L)-DIFF(L)*A13(L)
   33 RKSY(MA)=RY(L)
      CALL GTPRD(SZMOD,RKSY,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GTPRD(SYMOD,RZA11,GSMOD,NSTRUT,NMODE,NSTRUT)
      CALL GMSUB(CTMP1,GSMOD,GFORC2,NMODE,NSTRUT)
      CALL GTPRD(SXMOD,RZA11,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GTPRD(SZMOD,RXA13,GSMOD,NSTRUT,NMODE,NSTRUT)
      CALL GMSUB(CTMP1,GSMOD,GFORC3,NMODE,NSTRUT)
      CALL GTPRD(SYMOD,RXA13,CTMP1,NSTRUT,NMODE,NSTRUT)
      CALL GTPRD(SXMOD,RKSY,GSMOD,NSTRUT,NMODE,NSTRUT)
      CALL GMSUB(CTMP1,GSMOD,GFORC4,NMODE,NSTRUT)
C
```

```
C
C     FORM INITIAL DISPLACEMENTS
C
      SUM1=0.
      SUM3=0.
      DO 34 I=1,NSTRUT
      SUM1=SUM1+A31(I)*SDD((I-1)*2+1)
   34 SUM3=SUM3+A33(I)*SDD((I-1)*2+1)
      VARY1(1)=AX77F-SUM1
      VARY1(2)=AY77F
      VARY1(3)=AZ77F-SUM3
      CALL GMPRD(GFORC1,VARY1,QS,NMODE,3,1)
      DO 35 I=1,NSTRUT
      OMXD1M(I)=P177R1*MASS(I)
      OMYD1M(I)=Q177R1*MASS(I)
   35 OMZD1M(I)=R177R1*MASS(I)
      CALL GMPRD(GFORC2,OMXD1M,GF,NMODE,NSTRUT,1)
      CALL GMADD(QS,GF,QS,NMODE,1)
      CALL GMPRD(GFORC3,OMYD1M,GF,NMODE,NSTRUT,1)
      CALL GMADD(QS,GF,QS,NMODE,1)
      CALL GMPRD(GFORC4,OMZD1M,GF,NMODE,NSTRUT,1)
      CALL GMADD(QS,GF,QS,NMODE,1)
      CALL GTPRD(TXMOD,T,GF,IN,NMODE,1)
      CALL GMSUB(GF,QS,QS,NMODE,1)
      FOC(1)=FCX
      FOC(2)=FCY
      FOC(3)=FCZ
      CALL GTPRD(DCMODE,FOC,GF,3,NMODE,1)
      CALL GMADD(GF,QS,QS,NMODE,1)
      DO 36 I=1,NMODE
      NN=(I-1)*6+1
      CTMP1(NN)=-ARMODE(NN)*AA77P
      CTMP1(NN+1)=ARMODE(NN+1)*YA77P
      CTMP1(NN+2)=-ARMODE(NN+2)*ANA77P
      CTMP1(NN+3)=ARMODE(NN+3)*ALA77F
      CTMP1(NN+4)=ARMODE(NN+4)*AMA77F
   36 CTMP1(NN+5)=ARMODE(NN+5)*ANA77F
      DO 37 II=1,NMODE
      GF(II)=0.
      DO 37 JJ=1,6
      MJ=(II-1)*6+JJ
   37 GF(II)=GF(II)+CTMP1(MJ)*PF(MJ)
      CALL GMADD(GF,QS,QS,NMODE,1)
      DO 38 IG=1,NMODE
      GTF(IG)=GMASS1(IG)*GFREQ(IG)**2.
   38 GQ(IG)=QS(IG)/GTF(IG)
      SKIPUP=.TRUE.
      DO 39 I=1,NMODE
      N22=2*I
      CALL UPDAT(1,LA(N22),GQ(I),DU,DU,DU,DU)
   39 CONTINUE
      SKIPUP=.FALSE.
      RETURN
C
      ENTRY FLEX3
C
      IF(INOFLX.EQ.0.OR.INOLG.EQ.0)RETURN
C
C     FORM VARYING COMPONETS OF TANGENTIAL ACCELERATION
C
      SUM1=0.
      SUM3=0.
      DO 40 I= 1,NSTRUT
      SUM1=SUM1+A31(I)*SDD((I-1)*2+1)
   40 SUM3=SUM3+A33(I)*SDD((I-1)*2+1)
      VARY1(1)=AX77F-SUM1
```

```
      VARY1(2)=AY77F
      VARY1(3)=AZ77F-SUM3
      CALL GMPRD(GFORC1,VARY1,GTF,NMODE,3,1)
      DO 46 I=1,NMODE
   46 QS(I)=0.
      CALL GMSUB(QS,GTF,QS,NMODE,1)
      DO 41 I=1,NSTRUT
      OMXD1M(I)=PI77R1*MASS(I)
      OMYD1M(I)=QI77R1*MASS(I)
   41 OMZD1M(I)=RI77R1*MASS(I)
C
      CALL GMPRD(GFORC2,OMXD1M,GTF,NMODE,NSTRUT,1)
      CALL GMSUB(QS,GTF,QS,NMODE,1)
      CALL GMPRD(GFORC3,OMYD1M,GTF,NMODE,NSTRUT,1)
      CALL GMSUB(QS,GTF,QS,NMODE,1)
      CALL GMPRD(GFORC4,OMZD1M,GTF,NMODE,NSTRUT,1)
      CALL GMSUB(QS,GTF,QS,NMODE,1)
C
C     FORM GENERALIZED THRUST MATRIX
C
      CALL GTPRD(TXMOD,T,GTF,IN,NMODE,1)
      CALL GMADD(QS,GTF,QS,NMODE,1)
C
C     FORM GENERALIZED DRAG CHUTE FORCES
C
      FDC(1)=FCX
      FDC(2)=FCY
      FDC(3)=FCZ
      CALL GTPRD(DCMODE,FDC,GTF,3,NMODE,1)
      CALL GMADD(QS,GTF,QS,NMODE,1)
C
C     FORM GENERALIZED AERO FORCES
C
      DO 42 I=1,NMODE
      NN=(I-1)*6+1
      CTMP1(NN)=-ARMODE(NN)*AA77P
      CTMP1(NN+1)=ARMODE(NN+1)*YA77P
      CTMP1(NN+2)=-ARMODE(NN+2)*ANA77P
      CTMP1(NN+3)=ARMODE(NN+3)*ALA77F
      CTMP1(NN+4)=ARMODE(NN+4)*AMA77F
   42 CTMP1(NN+5)=ARMODE(NN+5)*ANA77F
      DO 43 II=1,NMODE
      GTF(II)=0.
      DO 43 JJ=1,6
      MJ=(II-1)*6+JJ
   43 GTF(II)=GTF(II)+CTMP1(MJ)*PF(MJ)
      CALL GMADD(QS,GTF,QS,NMODE,1)
C
C     FORM GENERALIZED STIFFNESS
C
      DO 44 IG=1,NMODE
      GTF(IG)=GQ(IG)*GMASS1(IG)*GFREQ(IG)**2.
   44 QS(IG)=QS(IG)-GTF(IG)
C
C     SOLVE FOR THE GENERALIZED ACCELERATION
C
      CALL SOLVE(NMODE,COEF,QS,GQD2)
      DO 45 I=1,NMODE
      NCON=(I-1)*2+1
      MCON=NCON+1
      CALL INTEG(LA(NCON),GQD2(I))
      CALL INTEG(LA(MCON),GQD1(I))
   45 CONTINUE
      RETURN
```

```
      ENTRY FLEX4
C
      RETURN
C
      ENTRY FLEX5
C
      RETURN
C
      ENTRY FLEX6
C
      IF(INDFLX.EQ.0.OR.INDLG.EQ.0)RETURN
      IF (NPTS.EQ.0) RETURN
C
C  FORM VARIABLES TO BE OUTPUT
C
      CALL STFL(0,1,HDR)
      CALL STFL(2,16,TITL)
      DO 62 I=1,NPTS
      DO 61 K=1,3
      SD0(K)=0.
      SD1(K)=0.
      SD2(K)=0.
      DO 61 J=1,NMODE
      NCON=((I-1)*3+K-1)*NMODE+J
      SD0(K)=SD0(K)+OUTMOD(NCON)*GQ(J)
      SD1(K)=SD1(K)+OUTMOD(NCON)*GQD1(J)
   61 SD2(K)=SD2(K)+OUTMOD(NCON)*GQD2(J)
      LX=(I-1)*3+1
      LY=LX+1
      LZ=LY+1
      XD2F(I)=SD2(1)
      XD2T(I)=AX77F+XD2F(I)+2.*(QI77R*SD1(3)-RI77R*SD1(2))+
     *(PI77R*RI77R+QI77R1)*(ROIS(LZ)+SD0(3))+(PI77R*QI77R
     *-RI77R1)*(ROIS(LY)+SD0(2))-(QI77R*QI77R+RI77R*RI77R)
     * *(ROIS(LX)+SD0(1))
      YD2F(I)=SD2(2)
      YD2T(I)=AY77F+YD2F(I)+2.*(RI77R*SD1(1)-PI77R*SD1(3))+
     *(QI77R*PI77R+RI77R1)*(ROIS(LX)+SD0(1))+(QI77R*RI77R
     *-PI77R1)*(ROIS(LZ)+SD0(3))-(RI77R*RI77R+PI77R*PI77R)
     * *(ROIS(LY)+SD0(2))
      ZD2F(I)=SD2(3)
      ZD2T(I)=AZ77F+ZD2F(I)+2.*(PI77R*SD1(2)-QI77R*SD1(1))+
     *(RI77R*QI77R+PI77R1)*(ROIS(LY)+SD0(2))+(RI77R*PI77R
     *-QI77R1)*(ROIS(LX)+SD0(1))-(PI77R*PI77R+QI77R*QI77R)
     * *(ROIS(LZ)+SD0(3))
      XD1F(I)=SD1(1)
      XD1T(I)=U777F+XD1F(I)+QI77R*(ROIS(LZ)+SD0(3))-RI77R*
     *(ROIS(LY)+SD0(2))
      YD1F(I)=SD1(2)
      YD1T(I)=V777F+YD1F(I)+RI77R*(ROIS(LX)+SD0(1))-PI77R*
     *(ROIS(LZ)+SD0(3))
      ZD1F(I)=SD1(3)
      ZD1T(I)=W777F+ZD1F(I)+PI77R*(ROIS(LY)+SD0(2))-QI77R*
     *(ROIS(LX)+SD0(1))
      XD0F(I)=SD0(1)
      YD0F(I)=SD0(2)
      ZD0F(I)=SD0(3)
      PTN=FLOAT(I)
      CALL STOVAR(8,PTN,XD2F(I),XD2T(I),YD2F(I),YD2T(I),ZD2F(I),
     *ZD2T(I),XD1F(I))
      CALL STOVAR(8,XD1T(I),YD1F(I),YD1T(I),ZD1F(I),ZD1T(I),XD0F(I),
     *YD0F(I),ZD0F(I))
   62 CONTINUE
      RETURN
C
```

```
C     ENTRY FLEX7

      IF(INDFLX.EQ.0.OR.INDLG.EQ.0)RETURN
      RVAR=2.*FLOAT(NMODE)/4.
      K=IFIX(RVAR)
      IF (K.EQ.0) GO TO 71
      DO 70 I=1,K
      N=4*I-3
      MD1=2*I-1
      MD0=2*I
      CALL UPDAT(4,LA(N),GQD1(MD1),GQ(MD1),GQD1(MD0),GQ(MD0),DU)
   70 CONTINUE
      L=2*K
      IF(L.EQ.NMODE)RETURN
   71 M=4*K+1
      MP=2*K+1
      CALL UPDAT(2,LA(M),GQD1(MP),GQ(MP),DU,DU,DU)
      RETURN
      END
      SUBROUTINE DECCMP (NN,A,UL)
      DIMENSION A(20,20), UL(20,20), SCALES(20), IPS(20)
      COMMON IPS
      IF (NN.GT.1) GO TO 20
      UL(1,1)=A(1,1)
      RETURN
   20 N=NN
C
C     INITIALIZE IPS, UL AND SCALES
      DO 5 I=1,N
      IPS(I)=I
      ROWNRM=0.0
      DO 2 J=1,N
      UL(I,J)=A(I,J)
      IF (ROWNRM-ABS(UL(I,J))) 1,2,2
    1 ROWNRM=ABS(UL(I,J))
    2 CONTINUE
      IF (ROWNRM) 3,4,3
    3 SCALES(I)=1.0/ROWNRM
      GO TO 5
    4 CALL SING(1)
      SCALES(I)=0.0
    5 CONTINUE
C
C     GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
      NM1=N-1
      DO 17 K=1,NM1
      BIG=0.0
      DO 11 I=K,N
      IP=IPS(I)
      SIZE=ABS(UL(IP,K))*SCALES(IP)
      IF (SIZE-BIG) 11,11,10
   10 BIG=SIZE
      IDXPIV=I
   11 CONTINUE
      IF (BIG) 13,12,13
   12 CALL SING(2)
      GO TO 17
   13 IF (IDXPIV-K) 14,15,14
   14 J=IPS(K)
      IPS(K)=IPS(IDXPIV)
      IPS(IDXPIV)=J
   15 KP=IPS(K)
      PIVOT=UL(KP,K)
      KP1=K+1
      DO 16 I=KP1,N
```

```
      IP=IPS(I)
      EM=-UL(IP,K)/PIVOT
      UL(IP,K)=-EM
      DO 16 J=KP1,N
      UL(IP,J)=UL(IP,J)+EM*UL(KP,J)
   16 CONTINUE
   17 CONTINUE
      KP=IPS(N)
      IF (UL(KP,N)) 19,18,19
   18 CALL SING(2)
   19 RETURN
      END
      SUBROUTINE SOLVE (NN,UL,B,X)
      DIMENSION UL(20,20), B(20), X(20), IPS(20)
      COMMON IPS
      IF (NN.GT.1) GO TO 5
      X(1)=B(1)/UL(1,1)
      RETURN
    5 N=NN
      NP1=N+1
C
      IP=IPS(1)
      X(1)=B(IP)
      DO 2 I=2,N
      IP=IPS(I)
      IM1=I-1
      SUM=0.0
      DO 1 J=1,IM1
    1 SUM=SUM+UL(IP,J)*X(J)
    2 X(I)=B(IP)-SUM
C
      IP=IPS(N)
      X(N)=X(N)/UL(IP,N)
      DO 4 IBACK=2,N
      I=NP1-IBACK
C        I GOES (N-1),.....,1
      IP=IPS(I)
      IP1=I+1
      SUM=0.0
      DO 3 J=IP1,N
    3 SUM=SUM+UL(IP,J)*X(J)
    4 X(I)=(X(I)-SUM)/UL(IP,I)
      RETURN
      END
      SUBROUTINE SING (IWHY)
   11 FORMAT(54H MATRIX WITH ZERO ROW IN DECOMPOSE.                   )
   12 FORMAT(54H SINGULAR MATRIX IN DECOMPOSE. ZERO DIVIDE IN SOLVE.    )
      NOUT=6
C       NOUT=STANDARD OUTPUT UNIT
      GO TO (1,2),IWHY
    1 WRITE (NOUT,11)
      GO TO 10
    2 WRITE (NOUT,12)
   10 CALL EXERR(0)
      RETURN
      END
      SUBROUTINE ARAY (MODE,I,J,N,M,S,D)
C          IDENT   ARAY
C          TITLE   ARAY
***        ADAPTED FROM S/360 SCIENTIFIC SUBROUTINE PACKAGE.
*           (360A-CM-03X) VERSION III
*           R. A. GARMOE.    08/16/69.
***
*
*     ..................................................................
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C        SUBROUTINE ARAY
C
C        PURPOSE
C           CONVERT DATA ARAY FROM SINGLE TO DOUBLE DIMENSION OR VICE
C           VERSA.  THIS SUBROUTINE IS USED TO LINK THE USER PROGRAM
C           WHICH HAS DOUBLE DIMENSION ARAYS AND THE SSP SUBROUTINES
C           WHICH OPERATE ON ARAYS OF DATA IN A VECTOR FASHION.
C
C     ..................................................................
C
      DIMENSION S(1),D(1)
      IF (I.GT.1) GO TO 98
      IF (MODE.EQ.1) D(1)=S(1)
      IF (MODE.EQ.2) S(1)=D(1)
      RETURN
C
   98 NI=N-I
C
C
C        TEST TYPE OF CONVERSION
C
      IF(MODE-1) 100, 100, 120
C
C        CONVERT FROM SINGLE TO DOUBLE DIMENSION
C
  100 IJ=I*J+1
      NM=N*J+1
      DO 110 K=1,J
      NM=NM-NI
      DO 110 L=1,I
      IJ=IJ-1
      NM=NM-1
  110 D(NM)=S(IJ)
      GO TO 140
C
C        CONVERT FROM DOUBLE TO SINGLE DIMENSION
C
  120 IJ=0
      NM=0
      DO 130 K=1,J
      DO 125 L=1,I
      IJ=IJ+1
      NM=NM+1
  125 S(IJ)=D(NM)
  130 NM=NM+NI
C
  140 RETURN
      END
      SUBROUTINE CTIE(A,B,R,N,M,MSA,MSB,L)
C           IDENT   CTIE
C           TITLE   CTIE
C**         ADAPTED FROM S/360 SCIENTIFIC SUBROUTINE PACKAGE.
C           (360A-CM-03X) VERSION III
C           R. A. GARMOE.    08/16/69.
C**
C     ..................................................................
C
C        SUBROUTINE CTIE
C
C        PURPOSE
C           ADJOIN TWO MATRICES WITH SAME ROW DIMENSION TO FORM ONE
C           RESULTANT MATRIX (SEE METHOD)
C
C     ..................................................................
C
```

```
*         DIMENSION A(1),B(1),R(1)
          MM=M
          IR=0
          MSX=MSA
          DO 6 JJ=1,2
          DO 5 J=1,MM
          DO 5 I=1,N
          IR=IR+1
          R(IR)=0.0
*
*            LOCATE ELEMENT FOR ANY MATRIX STORAGE MODE
*
          CALL LOC(I,J,IJ,N,MM,MSX)
*
*            TEST FOR ZERO ELEMENT IN DIAGONAL MATRIX
*
          IF(IJ) 2,5,2
*
*            MOVE ELEMENT TO MATRIX R
*
        2 GO TO(3,4),JJ
        3 R(IR)=A(IJ)
          GO TO 5
        4 R(IR)=B(IJ)
        5 CONTINUE
*
*            REPEAT ABOVE FOR MATRIX B
*
          MSX=MSB
          MM=L
        6 CONTINUE
          RETURN
          END
          SUBROUTINE LOC(I,J,IR,N,M,MS)
C               IDENT   LOC
C               TITLE   LOC
***              ADAPTED FROM S/360 SCIENTIFIC SUBROUTINE PACKAGE.
*                 (360A-CM-03X) VERSION III
*                 R. A. GARMOE.    08/16/69.

***
*         ..............................................................
*
*            SUBROUTINE LOC
*
*            PURPOSE
*               COMPUTE A VECTOR SUBSCRIPT FOR AN ELEMENT IN A MATRIX OF
*               SPECIFIED STORAGE MODE
*
*         ..............................................................
*
          IX=I
          JX=J
          IF(MS-1) 10,20,30
       10 IRX=N*(JX-1)+IX
          GO TO 36
       20 IF(IX-JX) 22,24,24
       22 IRX=IX+(JX*JX-JX)/2
          GO TO 36
       24 IRX=JX+(IX*IX-IX)/2
          GO TO 36
       30 IRX=0
```

```
      IF(IX-JX) 36,32,36
   32 IRX=IX
   36 IR=IRX
      RETURN
      END
      SUBROUTINE GTPRD(A,B,R,N,M,L)
C         IDENT   GTPRD
C         TITLE   GTPRD
C**        ADAPTED FROM S/360 SCIENTIFIC SUBROUTINE PACKAGE.
C           (360A-CM-03X) VERSION III
C           R. A. GARMOE.    08/16/69.
C**
C
C     ..................................................................
C
C        SUBROUTINE GTPRD
C
C        PURPOSE
C           PREMULTIPLY A GENERAL MATRIX BY THE TRANSPOSE OF ANOTHER
C           GENERAL MATRIX
C
C     ..................................................................
C
      DIMENSION A(1),B(1),R(1)
C
      IR=0
      IK=-N
      DO 10 K=1,L
      IJ=0
      IK=IK+N
      DO 10 J=1,M
      IB=IK
      IR=IR+1
      R(IR)=0
      DO 10 I=1,N
      IJ=IJ+1
      IB=IB+1
   10 R(IR)=R(IR)+A(IJ)*B(IB)
      RETURN
      END
      SUBROUTINE GMSUB(A,B,R,N,M)
C         IDENT   GMSUB
C         TITLE   GMSUB
C**        ADAPTED FROM S/360 SCIENTIFIC SUBROUTINE PACKAGE.
C           (360A-CM-03X) VERSION III
C           R. A. GARMOE.    08/16/69.
C**
C
C     ..................................................................
C
C        SUBROUTINE GMSUB
C
C        PURPOSE
C           SUBTRACT ONE GENERAL MATRIX FROM ANOTHER TO FORM RESULTANT
C           MATRIX
C
C     ..................................................................
C
      DIMENSION A(1),B(1),R(1)
C
C        CALCULATE NUMBER OF ELEMENTS
C
      NM=N*M
C
C        SUBTRACT MATRICES
C
      DO 10 I=1,NM
```

```
   10 R(I)=A(I)-B(I)
      RETURN
      END
      SUBROUTINE GMADD(A,B,R,N,M)
C           IDENT  GMADD
C           TITLE  GMADD
C***      ADAPTED FROM S/360 SCIENTIFIC SUBROUTINE PACKAGE.
*          (360A-CM-03X) VERSION III
*          R. A. GARMOE.    08/16/69.
C***
*
*     ..............................................................
*
*        SUBROUTINE GMADD
*
*        PURPOSE
*           ADD TWO GENERAL MATRICES TO FORM RESULTANT GENERAL MATRIX
*
*     ..............................................................
*
      DIMENSION A(1),B(1),R(1)
*
*        CALCULATE NUMBER OF ELEMENTS
*
      NM=N*M
*
*        ADD MATRICES
*
      DO 10 I=1,NM
   10 R(I)=A(I)+B(I)
      RETURN
      END
      SUBROUTINE GMPRD(A,B,R,N,M,L)
C           IDENT  GMPRD
C           TITLE  GMPRD
C***      ADAPTED FROM S/360 SCIENTIFIC SUBROUTINE PACKAGE.
*          (360A-CM-03X) VERSION III
*          R. A. GARMOE.    08/16/69.
C***
*
*     ..............................................................
*
*        SUBROUTINE GMPRD
*
*        PURPOSE
*           MULTIPLY TWO GENERAL MATRICES TO FORM A RESULTANT GENERAL
*           MATRIX
*
*     ..............................................................
*
      DIMENSION A(1),B(1),R(1)
*
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
   10 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE SDFLGP
      COMMON/DIRCOM/DM1(74),TIMER,DM2(61),AMASS,DM20(27),AX77F,DM21(4),
     *AZ77F,DM22(114),DCL1,DCM1,
     *DCN1,DCL2,DCM2,DCN2,DCL3,DCM3,DCN3,DM3(99),FX87P,
     *DUM4(3),FY87P,DM23(3),FZ87P,DM5(17),GX87F,DM6(8),
     *GZ87F,DM24(6),HGC7F,DM25(211),INDSTE,DM12(47),PHIPD,DM26(23),
     *PSIPD,
     *DM27(156),THTPD,DM28(5),TIME,DM29(12),TMAX,DMA(274),
     *PI77R,DM30,PI77R1,DM9(5),QI77R,DM31,QI77R1,DM10(5),
     *RI77R,DM32,RI77R1,DM11(49),XG77F,DM33,XG77F1,
     *DM34(11),YG77F,DM35,YG77F1,DM36(11),
     *ZG77F,DM37,ZG77F1,DUM13(53),
     CNSTRUT,MASS(5),RX(5),RY(5),RZ(5),THETAD(5),ERDEG,RGR,
     CNTIRES(5),RZERO(5),W(5),DELTAM(5),MOMENT(5),
     CRF(5),VZ   ,IFD,PZERO(5),VZERO(5),A(5),P22(5),V20(5),
     CA2(5),IL,S2T(5),ES2(5),C2L(5),MASS2(5),MUS(5),
     CCC(5),CE(5),C2C(5),C2E(5),NVGPT,NPP,MB(5),RLT,NDELTA,
     CE3(5),SB(5),SD21(2),SD22(2),SD23(2),SD24(2),SD25(2)
      COMMON/DIRCOM/
     CSD11(2),SD12(2),SD13(2),SD14(2),SD15(2),
     CS1(2),S52(2),S3(2),S4(2),S5(2),
     CS2021(2),S2022(2),S2023(2),S2D24(2),S2D25(2),
     CS2011(2),S2012(2),S2013(2),S2014(2),S2015(2),
     CS21(2),S22(2),S23(2),S24(2),S25(2),
     COMTD11(2),OMTD12(2),OMTD13(2),OMTD14(2),OMTD15(2),
     COMT1(2),OMT2(2),OMT3(2),OMT4(2),OMT5(2),
     CAI(5),BI(5),DELTA1,DELTA2,DELTA3,DELTA4,DELTAS,
     CDDELT1,DDELT2,DDELT3,DDELT4,DDFLT5,ISTAGE,
     CPRTMIN,IPLT,ISDF,ISTPL1,ISTPL2,ISTPL3,ISTPL4,ISTPL5,
     COM14(154),INDLG,DM15(151),DM15(607),NPTS,DPM17(1446),IFLX(20)
      COMMON/FLXOP/DDM17(338),XD2F(20),XD2T(20),YD2F(20),YD2T(20),
     *ZD2F(20),ZD2T(20),XD1F(20),XD1T(20),YD1F(20),YD1T(20),
     *ZD1F(20),ZD1T(20),XD0F(20),YD0F(20),ZD0F(20)
      REAL MASS,MOMENT,    MASS2,MUS,NTIRES,MB
      COMMON/LGDE/LA(25),FC2(5),P2(5),PRES(5),C(5),IPPT,LTPT
      COMMON/LG/FXP,FYM,FZM,LM,MM,NM
      COMMON/LGE/A11(5),A13(5),A31(5),A33(5),RRCGX(5),
     CRL(3,3),RI(3,3,5),RAX(5),RAY(5),RAZ(5),TMP(3),ZZERO(5),
     CXR(5),YR(5),EPSLON(5),PA(5),FDELTA(5),
     CFTRZ(5),RDX(5),RDY(5),RDZ(5),RDXG(5),RDYG(5),RDZG(5),
     CVTX(5),VTY(5),VTZ(5),GZ(5),          VGPT(5),FTRX(5),FTRY(5),
     CDX(5),DY(5),DZ(5),FT(5),FDX(5),FDY(5),FF(5),AA(5),C2(5),
     CSR(5),SF(5),PSKD(5),MUVP(5),MTRX(5),MTRY(5),
     CMTRZ(5),MA(5),RG11,RG13,RG31,RG33,IPRT,
     CMTX,MTY,MTZ,SFTRX,SFTRY,SFTRZ,FTRA,
     CFTRB,FTRC,SMTRX,SMTRY,SMTRZ
      REAL MUVP,MTRX,MTRY,MTRZ,MA,
     CMTX,MTY,MTZ,LM,MM,NM
      DIMENSION DELTA(5),DDELTA(5),P(5),
     C        SD2(2,5),SD1(2,5),S(2,5),S2D2(2,5),S2D1(2,5),S2(2,5),
     COMETD1(2,5),OMET(2,5)
      EQUIVALENCE  (P(1),PRES(1))
      EQUIVALENCE  (DELTA(1),DELTA1),(DDELTA(1),DDELT1)
      EQUIVALENCE
     C(SD21(1),SD2(1,1)),(SD11(1),SD1(1,1)),(S1(1),S(1,1)),
     C(S2D21(1),S2D2(1,1)),(S2D11(1),S2D1(1,1)),(S21(1),S2(1,1)),
     C(OMTD11(1),OMETD1(1,1)),(OMT1(1),OMET(1,1))
      DIMENSION OP16(8),OP17(8),OP18(8),OP19(4),OP20(8),OP21(8),
     *DAT2(15),DAT3(6),DAT4(15)
      DATA
     *(OP16(1),I=1,8)/5HDELTA,1HP,2HF2,2HFT,2HSR,2HSF,2HAA,3HFC2/,
     *(OP17(I),I=1,8)/3HSD2,3HSD1,1HS,4HS2D2,4HS2D1,2HS2,
     *6HOMETD1,4HOMET/,
     *(OP18(I),I=1,8)/4HFTRA,4HFTRB,4HFTRC,3HMTX,3HMTY,
     *3HMTZ,3HFXM,3HFYM/,
```

```
      *(OP19(I),I=1,4)/3HFZM,2HLM,2HMM,2HNM/,
      *(OP20(I),I=1,8)/4HMOVP,4HVGPY,4HFTRX,4HFTRY,4HFTRZ,
      *2HMA,2HMA),6HDELTA/,
      *(OP21(I),I=1,8)/5HDELTA,2HFT,4HMUVP,4HFTRX,4HFTRY,
      *4HFTRZ,6H)DELTA,1HS/
       DATA HBCI/5HLGCAR /
       DATA DAT1/4HTIME/,(DAT2(I),I=1,15)/2HLM,2HMM,2HNM,5HQI77R,
      *5HTHTPD,5HPSIPD,5HPHIPD,3HFZM,5HAZ77F,6HZG77F1,5HHGC7F,
      *5HXG77F,5HYG77F,5HAX77F,6HXG77F1/,
      *(DAT3(I),I=1,6)/2HFT,2HSF,5HDELTA,1HP,2HP2,2HMA/,
      *(DAT4(I),I=1,15)/4HXD2F,4HXD2T,4HYD2F,4HYD2T,4HZD2F,4HZD2T,
      *4HXD1F,4HXD1T,4HYD1F,4HYD1T,4HZD1F,4HZD1T,4HXD0F,4HYD0F,4HZD0F/
       DATA DAT5/6HLASTPT/,N1/1/,N15/15/,N14/14/
C      SAVE DATA ON TAPE
       IF(IPLT.EQ.0)RETURN
       LASTPT=0
       IF(IPLT.GT.1) GO TO 201
       NHL=2
       IF(ISDF.NE.0)NHL=NHL+1
       ISUM1=ISTPL1+ISTPL2+ISTPL3+ISTPL4+ISTPL5
       IF(ISUM1.NE.0)NHL=NHL+1
       ISUM2=0
       DO 105 I=1,NPTS
   105 ISUM2=ISUM2+IFLX(I)
       IF(ISUM2.NE.0)NHL=NHL+1
       WRITE(13)NHL
       WRITE(13)N1,N1,DAT1
       IF(ISDF.NE.0)WRITE(13)N15,N1,DAT2
       IF(ISUM1.NE.0)WRITE(13)N14,ISUM1,DAT3,OP17
       IF(ISUM2.NE.0)WRITE(13)N15,ISUM2,DAT4
       WRITE(13)N1,N1,DAT5
       IPLT=IPLT+1
   201 WRITE(13) TIME
       IF(ISDF.NE.0) WRITE(13) LM,MM,NM,QI77R,THTPD,PSIPD,PHIPD,FZM,
      *AZ77F,ZG77F1,HGC7F,XG77F,YG77F,AX77F,XG77F1
       IF(ISTPL1.NE.0) WRITE(13) FT(1),SF(1),DELTA(1),P(1),P2(1),MA(1),
      *SD2(1,1),SD1(1,1),S(1,1),S2D2(1,1),S2D1(1,1),S2(1,1),OMETD1(1,1),
      *OMET(1,1)
       IF(ISTPL2.NE.0) WRITE(13) FT(2),SF(2),DELTA(2),P(2),P2(2),MA(2),
      *SD2(1,2),SD1(1,2),S(1,2),S2D2(1,2),S2D1(1,2),S2(1,2),OMETD1(1,2),
      *OMET(1,2)
       IF(ISTPL3.NE.0) WRITE(13) FT(3),SF(3),DELTA(3),P(3),P2(3),MA(3),
      *SD2(1,3),SD1(1,3),S(1,3),S2D2(1,3),S2D1(1,3),S2(1,3),OMETD1(1,3),
      *OMET(1,3)
       IF(ISTPL4.NE.0) WRITE(13) FT(4),SF(4),DELTA(4),P(4),P2(4),MA(4),
      *SD2(1,4),SD1(1,4),S(1,4),S2D2(1,4),S2D1(1,4),S2(1,4),OMETD1(1,4),
      *OMET(1,4)
       IF(ISTPL5.NE.0) WRITE(13) FT(5),SF(5),DELTA(5),P(5),P2(5),MA(5),
      *SD2(1,5),SD1(1,5),S(1,5),S2D2(1,5),S2D1(1,5),S2(1,5),OMETD1(1,5),
      *OMET(1,5)
       DO 17 I=1,NPTS
       IF(IFLX(I).NE.0) WRITE(13) XD2F(I),XD2T(I),YD2F(I),YD2T(I),
      *ZD2F(I),ZD2T(I),XD1F(I),XD1T(I),YD1F(I),YD1T(I),ZD1F(I),ZD1T(I),
      *XD0F(I),YD0F(I),ZD0F(I)
    17 CONTINUE
       GO TO 24
       ENTRY LGCAUP
       IF(IPLT.EQ.0)RETURN
       IF(LTPT.EQ.0)RETURN
       LTPT=0
       LASTPT=1
       BACKSPACE 13
    24 WRITE(13)LASTPT
       IF(LASTPT.EQ.0)RETURN
       END FILE 13
       RETURN
       ENTRY LGCAR6
C      TIME HISTORY PRINT
       IF(INDLG.EQ.0)RETURN
       CALL STFL(3,1,HBCI)
```

```
      IF(INDLG.EQ.2)GO TO 30
      CALL STFL(2,8,OP16)
      DO 121 I=1,NSTRUT
      CALL STOVAR(8,DELTA(I),P(I),P2(I),FT(I),SR(I),SF(I),AA(I),FG2(I))
121   CONTINUE
      CALL STFL(2,8,OP20)
      DO 123 I=1,NSTRUT
      CALL STOVAR(8,MUVP(I),VGPT(I),FTRX(I),FTRY(I),FTRZ(I),
     *MA(I),MB(I),DDELTA(I))
123   CONTINUE
      CALL STFL(2,8,OP17)
      DO 122 I=1,NSTRUT
      CALL STOVAR(8,SD2(1,I),SD1(1,I),S(1,I),S2D2(1,I),
     CS2D1(1,I),S2(1,I),OMETD1(1,I),OMET(1,I))
122   CONTINUE
      GO TO 32
30    CALL STFL(2,8,OP21)
      DO 31 I=1,NSTRUT
      CALL STOVAR(8,DELTA(I),FT(I),MUVP(I),FTRX(I),FTRY(I),
     CFTRZ(I),DDELTA(I),S(1,I))
31    CONTINUE
32    CALL STFL(2,8,OP18)
      CALL STOVAR(8,FTRA,FTRB,FTRC,MTX,MTY,MTZ,
     CFXM,FYM)
      CALL STFL(2,4,OP19)
      CALL STOVAR(4,FZM,LM,MM,NM,DU,DU,DU,DU)
      RETURN
C     **********
      ENTRY LGEAR5
C     **********
C     COMPUTE AND PRINT CODES
      RETURN
C
      ENTRY LGEAR7
C
      IF(INDLG.EQ.0)RETURN
      IF(INDLG.EQ.2)RETURN
      DO  6 I=1,NSTRUT
      IF(IL.NE.3)GO TO 5
      I1=2*I+3*NSTRUT-1
      CALL UPDAT(2,LA(I1),S2D1(1,I),S2(1,I),DU,DU,DU)
5     I1=2*I+NSTRUT-1
      CALL UPDAT(2,LA(I1),SD1(1,I),S(1,I),DU,DU,DU)
      CALL UPDAT(1,LA(I),OMET(1,I),DU,DU,DU,DU)
6     CONTINUE
      RETURN
      END
      OVERLAY(TOLA,1,0)
      PROGRAM TOLAN1
      COMMON/CONTRO/ICONTR,CONTR1,ICONT2,ICONT3
      IF(ICONTR-2)1,2,3
1     CALL READ
      GO TO 19
2     CALL TFFS2
      GO TO 19
3     IF(ICONTR-4)4,5,6
4     CALL SAC32
      GO TO 19
5     CALL VPCTAB
      GO TO 19
6     IF(ICONTR-6)7,8,9
7     CALL OPTTAB
      GO TO 19
8     CALL LGTAB
      GO TO 19
```

```
      CALL DJERGH(CONTR1,ICONT2,ICONT3)
 19   CONTINUE
      END
      SUBROUTINE TFFS2
      COMMON/TABSRC/LOCTFF(2),LOCSAC(79),LOCVPC(18),LOCSDF(4),LOCLG(7)
C
C     SETTING UP TABLES FOR TFFS
C
      CALL TSRCH(6HTTAB1.,LOCTFF,2,IER)
      RETURN
C
C     SETTING UP TABLES FOR SACS
C
      ENTRY SAC2
      CALL TSRCH(6HATAB01,LOCSAC(1),79,IER)
      RETURN
C
C     SETTING UP TABLES FOR VPCS2
C
      ENTRY VPCTAB
      CALL TSRCH(6HVTAB01,LOCVPC(1),18,IER)
      RETURN
C
C     SETTING UP TABLES FOR OPT2
C
      ENTRY OPTTAB
      CALL TSRCH(6HOTAB01,LOCSDF(1),4,IER)
      RETURN
C
C     SETTING UP TABLES FOR LGEAR
C
      ENTRY LGTAB
      CALL TSRCH(6HFTAB01,LOCLG(1),3,IER)
      CALL TSRCH(6HCTAB01,LOCLG(4),4,IER)
      RETURN
      END
      SUBROUTINE READ
      COMMON /READ1/ FI(56) , ITABLE , IID , IOD,
     *IBC,K,IBCRW,INX,SLTSYM,JBC,INXO
      DIMENSION MSG(58),RA(55),RA1(6)
      EQUIVALENCE (MSG( 1),SYM) , (MSG( 2),OP )
      EQUIVALENCE (MSG( 3),RA ) , (MSG(58),INC)
      REAL FW(2,28),MRG,NOKEQ
      LOGICAL STSRCH , TABSTP
      COMMON /STOPIT/ STSRCH , TABSTP
      DATA BCD,AINT,STCASE,TRA,MRG,NOKEQ,OCT,COMMA/
     *     3HBCD,3HINT,6HSTCASE,3HTRA,3HMRG,1HN,3HOCT,1H,/
      DATA BLANK,POINT,EE/1H ,1H.,1HE/
      DATA TABLED / 3HTAB /
    1 FORMAT (11E12.0)
    2 FORMAT (5X,I6,10E12.0)
    3 FORMAT (11I12)
    4 FORMAT(A6,1X,A3,1X,55A1,6A1)
    5 FORMAT (23H ERROR.THE SYMBOL **A6,26H** IS NOT IN THE DIRECTORY/
     *       1H )
    6 FORMAT (22A6)
    7 FORMAT (  I6)
    8 FORMAT (40H ERROR.AN STCASE MRG CARD HAS BEEN ENCOUNTERED,
     *       42H FOR WHICH NO BASE CASE IS AVAILABLE.ERROR    )
    9 FORMAT (52H ERROR.INPUT DECK NOT STARTED WITH STCASE CARD.ERROR)
   10 FORMAT (18X,A6,1X,A3,1X,55A1,I6)
   11 FORMAT (25H ERROR.COLUMN 12 IS BLANK/1H )
   12 FORMAT (27H NO-EXECUTE OPTION SELECTED)
   13 FORMAT (11I12)
   14 FORMAT (55A1)
```

```
      FORMAT (I1 , 9A6)
      K = 0
      DO 16 I=1,2
      DO 16 J=1,28
16    FU(I,J)=0.
100   READ(5,4)SYM,OP,RA,RA1
      IF(EOF,5) 18,19
18    REWIND 13
      STOP 22
19    CALL DIPLAC(RA1,INC,BLANK)
C
20    IF (SYM .NE. STCASE) GO TO 150
21    STSWCH = .FALSE.
      CALL DEF
101   IF (OP  .NE. MRG   ) GO TO 115
      IF (JBC .LT.0) GO TO 700
      IF (IBC .EQ.0) GO TO 110
C TERMINATE BASE CASE AUXILIARY FILE
      IID = -10
      WRITE (16) IID,IID,FI,SYM,OP,RA,INC,ITABLE
      ENDFILE 16
110   IBC = 0
      REWIND 16
      IBCRW= 0
      K = 1
      GO TO 120
115   IBC = 1
      INXQ= 1
      JBC = 1
      K    = 1
      IF (OP .NE. TABLED) GO TO 120
      CALL TABRE
      IF (TABSTP) GO TO 306
120   INX = 1
      IF ( RA(1) .NE. NOXEQ) GO TO 124
      INX = 0
      CALL LINES (2)
      WRITE (6,12)
124   IF ( IBC .EQ. 0) GO TO 130
      REWIND 16
      IBCRW = 0
      GO TO 130
130   IF (INX .EQ. 0) GO TO 801
      IF (IBC .EQ. 0)CALL READA (IBCRW)
      GO TO 100
150   IF (JBC .NE. 1) GO TO 705
      IF (STSWCH)       GO TO 306
      IF (K    .EQ. 0) GO TO 170
152   IF (OP   .NE. TRA) GO TO 200
      CALL LINES (1)
      WRITE (6,10) SYM,OP
      IF (IBC .EQ. 0) GO TO 160
      IID = -1
      WRITE (16) IID,IID,FI,SYM,OP,RA,INC,ITABLE
160   IF (INX .EQ. 0) GO TO 100
      IF (INXQ.NE. 1) GO TO 100
      SLTSYM = SYM
      RETURN
170   IF (IBC .NE.0) GO TO 152
175   CALL READA (IBCRW)
      K = 1
      GO TO 152
200   IF (K   .NE. 0) GO TO 300
      IF (INX.EQ. 0) GO TO 295
[illegible]   IF (IBC.EQ. 0) CALL READA (IBCRW)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
305   K = 1
300   CALL WRCARD (MSG)
301   IF (RA(1) .NE. BLANK) GO TO 302
24    CALL LINES (2)
      WRITE (6,11)
      GO TO 301
302   IF (SYM .EQ. SLTSYM) GO TO 310
      CALL DSERCH (SYM,IDD,ITABLE)
      IF (ITABLE .LT. 1) GO TO 310
      CALL LINES (2)
      WRITE (6,5) SYM
306   IF (IBC.NE. 1) GO TO 307
      INXQ = 0
      JBC  =-1
307   INX   = 0
      GO TO 801
C
C DECODE CARD SECTION
C
 310  IF (OP .NE. BCD) GO TO 400
C
C DATA IS BCD
C
      DECODE(1,1001,RA)N
 1001 FORMAT(I1)
      K1=2
      KK=7
      DO 1088 I=1,N
      ENCODE(10,888,FI(I))(RA(L),L=K1,KK)
      K1=K1+6
      KK=KK+6
 1088 CONTINUE
 888  FORMAT(5A1)
      CALL STORE(N,INC,INX,L)
      IF (IBC .EQ. 0) GO TO 350
      WRITE (16) IID,N,FI,SYM,OP,RA,INC   ,ITABLE
 350  SLTSYM = SYM
      GO TO 100
C
C NUMERIC TYPE DATA
C
 400  IFI = 0
C OCTAL TYPE
      IF (OP .EQ. OCT) IFI=1
C INTEGER TYPE
      IF (OP .EQ.AINT) IFI=4
 405  IQ=0
      LQ=0
      JJ=1
 410  KQ = LQ + 1
 415  LQ = LQ+1
      IF (LQ .GT. 55) GO TO 418
      IF (RA(LQ) .EQ. COMMA) GO TO 420
      IF (RA(LQ) .NE. BLANK) GO TO 415
 418  IQ=1
 420  MQ=LQ-1
      IF (IFI .NE. 0) GO TO 500
 425  JQ=KQ-1
 430  JQ=JQ+1
      IF (RA(JQ) .EQ. POINT) GO TO 435
      IF (RA(JQ) .NE. EE    ) GO TO 440
 435  IFI = 2
      GO TO 500
 440  IF ( JQ .LT. MQ) GO TO 430
      IFI = 3
```

```
  0J    CALL PACKRR(RA(KQ),E0(1,JJ),MQ-KQ+1)
        IF(IQ.NE.0)GO TO 525
  520   JJ=JJ+1
        GO TO 410
  525   CALL RITE(IFI,F0,FI,JJ)
  600   CALL STORE(JJ,INC,INX,0)
        IF (IBC .NE. 0) WRITE (16) IID,JJ,FI,SYM,OP,RA,INC,ITABLE
        GO TO 551
  700   CALL LINES (1)
        WRITE (6,8)
        GO TO 801
  705   CALL LINES (1)
        CALL DEF
        WRITE (6,9)
  801   SLTSYM = SYM
  802   READ (5,4) SYM,OP,RA,INC
        IF(EOF,5) 25,26
  25    REWIND 13
        STOP 25
  26    IF(SYM.EQ.STCASE)GO TO 21
        CALL WRCARD (MSG)
        IF (OP     .EQ. TRA    ) GO TO 802
        IF (RA(1)  .EQ. BLANK  ) GO TO 24
        IF (SYM    .EQ. SLTSYM) GO TO 802
        CALL DSERCH (SYM,IOD,ITABLE)
        IF (ITABLE .LT. 1) GO TO 801
        CALL LINES (1)
        WRITE (6,5) SYM
        GO TO 801
        END
         SUBROUTINE DIPLAC(RA1,INC,BLANK)
         DIMENSION RA1(6)
         DIMENSION RC(2)
C        ***************************************************************
CPICK UP COLUMN 67 TO 72 AND
CRIGHT JUSTIFY FOR USE IN THE INPUT ROUTINE
C
C
C
C        ***************************************************************
         DO 7 I=1,6
         IF (RA1(6)  .NE. BLANK) GO TO 2
         DO 1 N=1,6
         N1=7-N
         N2=6-N
  1      RA1(N1)=RA1(N2)
  7      RA1(1)=BLANK
         INC=0
C
C
C       IF EXIT THROUGH GO STATEMENT 2
C       THEN A VALID NUMBER EXIST.
C
         GO TO 6
  2     CALL PACKRR(RA1,RC,6)
        CALL RITE(3,RC,INC,1)
  6      RETURN
        END
        SUBROUTINE TABRE
        COMMON/TABDIR/TABLE(1)
        COMMON  /STOPIT/ STSWCH , TABSTP
        COMMON/TABCOM/LOCS(115),STABLE(115)
        DIMENSION IS(115),RA1(6)
        LOGICAL STSWCH , TABSTP
        DATA BLANK , BCDTRA / 1H ,3HTRA /
```

```
      DATA MAXT  , MAXTAB / 115,800/
      TABSTP = .FALSE.
      DO 5 I=1,MAXT
5     IS(I) = 0
      SYMTRA = BLANK
      NSUM= 0
      NTR=0
      WRITE (6,3)
3     FORMAT (35X,43HTABLE SIZES FOR EACH TABLE USED IN THIS RUN)
10    READ(5,11)SYM,SYMTRA,RA1
11    FORMAT(A6,1X,A3,1X,6A1)
      CALL DIPLAC(RA1,NTSIZE,BLANK)
      CALL LINES (1)
      IF (SYMTRA .EQ. BCUTRA) GO TO 21
      WRITE (6,4) SYM,SYMTRA,NTSIZE
4     FORMAT (13X,A6,1X,A3,1X,I4)
      DO 15 J=1,MAXT
      IF (SYM .NE. STABLE(J) ) GO TO 15
      IS(J)=NTSIZE
      NSUM = NSUM+NTSIZE
      NTR=NTR+1
      GO TO 10
15    CONTINUE
      TABSTP = .TRUE.
      CALL LINES (2)
      WRITE (6,24) SYM
24    FORMAT (1H0,36X,A6,29H DOES NOT EXIST IN TABLE LIST)
      GO TO 10
21    CALL LINES (2)
      WRITE (6,4 ) BLANK,SYMTRA
      NSUM=NSUM+MAXT-NTR
      WRITE (6,22) NSUM
22    FORMAT(35X,     39HTOTAL NUMBER OF CELLS FOR ALL TABLES =  I4)
      IF (NSUM .LE. MAXTAB) GO TO 200
      TABSTP = .TRUE.
      CALL LINES(2)
      WRITE(6,23) MAXTAB , NSUM
23    FORMAT(1H0,10X,25HTOTAL TABLE SIZES EXCEED I4,4H .  ,
     *35HCHANGE MAXIMUM TOTAL TABLE SIZES TO ,I4/
     *25H IN THE MAIN PROGRAM TOLA)
200   KONT=1
      DO 250 I=1,MAXT
      LOCS(I)=KONT
250   KONT = KONT+IS(I)
      RETURN
      END
      SUBROUTINE READA (IBCRW)
      COMMON/READ1/FI(56),ITABLE,IID,IOD,DM(7)
      INTEGER MSG(58)
      IF (IBCRW .NE. 0) RETURN
100   READ (16) IID,J ,FI,MSG , ITABLE
      IF (IID .GE. -1) GO TO 105
      IBCRW = 1
      RETURN
105   CALL WRCARD (MSG)
      IF (IID .LT. 0) RETURN
      CALL STORE(J,IDUM1,IDUM2,1)
      GO TO 100
      END
      SUBROUTINE STORE(N, INC, INX,STAPE)
      INTEGER STAPE
      COMMON/READ1/FI(56),ITABLE,IID,IOD,DM(7)
      COMMON/DIRCOM/DATA(3959)
      COMMON/TABDIR/TABLE(1)
      IF(STAPE.NE.0)GO TO 50
```

```
O      M1ORO = 0
       IF (INC.GT.0) M1ORO = -1
       IID = IDD+INC+M1ORO
       IF (INX .EQ. 0) RETURN
 50    J=IID
       IF (ITABLE .LT. 0) GO TO 2
       DO 1 I=1,N
       DATA (J) = FI(I)
 1     J = J+1
       RETURN
 2     DO 3 I=1,N
       TABLE(J) = FI(I)
 3     J = J+1
       RETURN
       END
       SUBROUTINE WRCARD(MSG)
       INTEGER MSG(58)
C
       CALL LINES (1)
       J = 58
       IF (MSG(58) .LE. 0) J =57
       WRITE (6,10) (MSG(I),I=1,J)
 10    FORMAT (18X,A6,1X,A3,1X,55A1,I6)
       RETURN
       END
       SUBROUTINE PACK (I1,I2,N)
       DIMENSION I1(1)
       DATA  IBL / 6H      /
     1 FORMAT (6A1)
     2 FORMAT ( A6)
       REWIND 31
       IF (N.GE.6) GO TO 3
       K = 6 - N
       WRITE (31,1) (IBL, I = 1,K),(I1(I), I = 1,N)
       GO TO 4
     3 WRITE (31,1) (I1(I), I = 1,6)
 4     REWIND 31
       READ (31,2) I2
       RETURN
       END
       SUBROUTINE DSERCH (SYM,IOC,IER)
       COMMON/FIXDIP/ANAME(900),LOC(900),NCOUNT
       DO 50 I=1,NCOUNT
       IF (SYM .NE. ANAME(I) ) GO TO 50
       IER = 0
       IOC = LOC(I)
       RETURN
 50    CONTINUE
       CALL TSRCH (SYM,IOC,1,IER)
       RETURN
C
       END
       SUBROUTINE PACKRR(I1,I2,NNN)
       DIMENSION I1(1)
 1000  FORMAT(1H(,I2,2HX,I2,3HA1))
       K=20-NNN
       ENCODE(10,1000,XMAT)K,NNN
       ENCODE(20,XMAT,I2)(I1(I),I=1,NNN)
       RETURN
       END
       SUBROUTINE RITE(IFI,FJ,FI,JJ)
       DIMENSION FI(56),FJ(56)
       GO TO (100,200,300,400),IFI
 100   II=1
       DO 101 I=1,JJ
```

```
      DECODE(20,1,FJ(II))FI(I)
      II=II+2
101   CONTINUE
1     FORMAT(O20)
      RETURN
200   II=1
      DO 201 I=1,JJ
      DECODE(20,2,FJ(II))FI(I)
      II=II+2
201   CONTINUE
2     FORMAT(E20.0)
      RETURN
300   II=1
      J1=JJ
      JJ=1
      DECODE(20,5,FJ)(FI(I),I=1,JJ)
      JJ=J1
      IF(JJ.EQ.1)RETURN
      II=II+2
      DO 301 I=2,JJ
      DECODE(20,2,FJ(II))FI(I)
      II=II+2
301   CONTINUE
5     FORMAT(10X,I10/(E20.0))
3     FORMAT(I20,(E20.0))
      RETURN
400   II=1
      DO 401 I=1,JJ
      DECODE(20,4,FJ(II))FI(I)
      II=II+2
401   CONTINUE
4     FORMAT(I20)
      RETURN
      END
      BLOCK DATA DIRDDA
      COMMON/FIXDIR/NAME(900),LOC(900),NCOUNT
C              GENERAL DIRECTORY ROUTINE
      DATA   (NAME(I),I=1,126)/
     *   6HAA77P ,6H-ELER1,6HAE77F ,6HAINCPS,6HAIXR7S,6HAIXXBS,6HAIXXS1,
     *   6HAIXYBS,6HAIXYS1,6HAIXZBS,6HAIXZS1,6HAIYYBS,6HAIYYS1,6HAIYZBS,
     *   6HAIYZS1,6HAIZZBS,6HAIZZS1,6HAJG   ,6HAKA   ,6HAKB   ,6HAKD   ,
     *   6HAKF   ,6HAKGPAV,6HAKZ   ,6HAK1   ,6HTR    ,6HTIMER ,6HAK13  ,
     *   6HAK14  ,6HAK15  ,6HAK16  ,6HAK17  ,6HAK18  ,6HAK19  ,6HAK2   ,
     *   6HAK20  ,6HAK21  ,6HAK22  ,6HAK23  ,6HAK24  ,6HAK25  ,6HAK26  ,
     *   6HAK3   ,6HAK6   ,6HAK7   ,6HALAMF1,6HALAMF2,6HALAMIF,6HALAMTD,
     *   6HALA77F,6HALB77F,6HALH77F,6HALIFTP,6HALIMG ,6HALIML ,6HALLJDF,
     *   6HALMJDF,6HALNJDF,6HALPHD ,6HALPHR1,6HALPNO ,6HALPTD ,6HALT77F,
     *   6HALYJDF,6HALZJDF,6HAL01  ,6HAL02  ,6HAL03  ,6HAL04  ,6HAMACH ,
     *   6HAMASFS,6HAMASF1,6HAMASS ,6HAMASS1,6HAMASZS,6HAMAXER,6HAMAXH ,
     *   6HAMA77F,6HAMB77F,6HAMINER,6HAMT77F,6HAMU   ,6HAMU77H,6HAN    ,
     *   6HANAB7G,6HANAZ7F,6HANA77P,6HANB77F,6HANGAMG,6HANPSIG,6HANSIGG,
     *   6HANTHTG,6HANT77F,6HANUA7F,6HANVW7G,6HAREFF ,6HAXP7F ,6HAX77F ,
     *   6HAYP7F ,6HAY77F ,6HAZP7F ,6HAZ77D ,6HAZ77F ,6HA777D ,6HBA77D ,
     *   6HBELER2,6HBETAD ,6HBETAR1,6HBI77D ,6HBLIMG ,6HBLIML ,6HBP77D ,
     *   6HB01   ,6HB02   ,6HB12   ,6HB13   ,6HB141  ,6HB142  ,6HB143  ,
     *   6HB144  ,6HB16   ,6HB17   ,6HB18   ,6HB19   ,6HB20   ,6HB21   /
      DATA   (NAME(J),J=127,252)/
     *   6HB22   ,6HCA    ,6HCALPG ,6HCAMNU ,6HCBETG ,6HCD    ,6HCDMNU ,
     *   6HCDYNP ,6HCGAMG ,6HCGAM71,6HCL    ,6HCLMNU ,6HCM    ,6HCMMNU ,
     *   6HCN2MNU,6HCN1MNU,6HCP    ,6HCRM   ,6HCY    ,6HCYA   ,6HCYM   ,
     *   6HCYMNU ,6HDECRES,6HDELNO ,6HDELPD ,6HDELQD ,6HDELRD ,6HDELTS ,
     *   6HDELT1 ,6HDELT2 ,6HDELT3 ,6HDLFXP ,6HDLFYP ,6HDLFZP ,6HDLH1  ,
     *   6HDLLTF ,6HDLMTF ,6HDLNTF ,6HDLTM1 ,6HDLTM2 ,6HDLTM3 ,6HDLTSF ,
     *   6HDRAGP ,6HDTC2R ,6HDXCGF ,6HDYNPP ,6HDYNPP1,6HD1    ,6HD1RFF ,
     *   6HD2    ,6HD2RFF ,6HD3    ,6HD4    ,6HD5    ,6HD6    ,6HD777D ,
```

```
     6HEPS1  ,6HEPS10 ,6HEPS11 ,6HEPS12 ,6HEPS13 ,6HEPS14 ,6HEPS15 ,
     6HEPS16 ,6HEPS17 ,6HEPS18 ,6HEPS19 ,6HEPS2  ,6HEPS20 ,6HEPS21 ,
     6HEPS22 ,6HEPS23 ,6HEPS24 ,6HEPS25 ,6HEPS26 ,6HEPS3  ,6HEPS4  ,
     6HEPS5  ,6HEPS6  ,6HEPS7  ,6HEPS8  ,6HEPS9  ,6HFXB7P ,6HFXE7P ,
     6HFYB7P ,6HFYE7P ,6HFZB7P ,6HFZE7P ,6HGAMDD ,6HGAMIF ,6HGAMIF1,
     6HGAMIF2,6HGAM7D ,6HGREFF ,6HGXB7F ,6HGXE7F ,6HGXG7F ,6HGXI7F ,
     6HGYB7F ,6HGYE7F ,6HGYI7F ,6HGZB7F ,6HGZE7F ,6HGZG7F ,6HGZI7F ,
     6HHA77D ,6HHG    ,6HHGC7F ,6HHH    ,6HHRFAB ,6HPRINT ,6HIA80X ,
     6HIA80Y ,6HIA81X ,6HIA81Y ,6HIA82X ,6HIA82Y ,6HIA83X ,6HIA83Y ,
     6HIA84X ,6HIA84Y ,6HIA85X ,6HIA85Y ,6HIA86X ,6HIA86Y ,6HIA87X /
      DATA   (LOC(I),I=1,126)/
     23     , 11     , 24     , 25     , 29     , 30     , 31     ,
     32     , 33     , 34     , 35     , 37     , 38     , 39     ,
     40     , 42     , 43     , 44     , 45     , 46     , 47     ,
     48     , 49     , 72     , 73     , 74     , 75     , 76     ,
     77     , 78     , 79     , 80     , 81     , 82     , 83     ,
     84     , 85     , 86     , 87     , 88     , 89     , 90     ,
     91     , 94     , 95     , 99     , 102    , 101    , 103    ,
     105    , 106    , 108    , 109    , 110    , 111    , 112    ,
     113    , 116    , 117    , 121    , 123    , 125    , 127    ,
     128    , 129    , 130    , 131    , 132    , 133    , 134    ,
     1138   ,1140    , 137    , 138    , 139    , 10     , 140    ,
     141    , 142    , 9      , 145    , 146    , 147    , 149    ,
     150    , 151    , 152    , 153    , 155    , 156    , 157    ,
     158    , 159    , 160    , 162    , 163    , 164    , 165    ,
     166    , 167    , 168    , 169    , 170    , 171    , 173    ,
     17     , 175    , 179    , 181    , 183    , 184    , 185    ,
     187    , 188    , 189    , 190    , 191    , 192    , 193    ,
     194    , 195    , 196    , 197    , 198    , 199    , 200    /
      DATA   (LOC(J),J=127,252)/
     201    , 202    , 852    , 205    , 863    , 212    , 215    ,
     216    , 859    , 860    , 222    , 226    , 227    , 228    ,
     229    , 230    , 231    , 242    , 250    , 250    , 252    ,
     253    , 321    , 325    , 327    , 329    , 331    , 333    ,
     334    , 335    , 336    , 337    , 338    , 339    , 340    ,
     341    , 343    , 344    , 345    , 346    , 347    , 348    ,
     349    , 350    , 351    , 355    , 353    , 356    , 357    ,
     358    , 359    , 360    , 361    , 362    , 363    , 144    ,
     366    , 367    , 368    , 369    , 370    , 371    , 372    ,
     373    , 374    , 375    , 376    , 377    , 378    , 379    ,
     380    , 381    , 382    , 383    , 384    , 385    , 386    ,
     387    , 388    , 389    , 390    , 391    , 393    , 395    ,
     397    , 349    , 401    , 403    , 406    , 410    , 411    ,
     412    , 414    , 417    , 419    , 420    , 421    , 422    ,
     424    , 425    , 426    , 428    , 429    , 430    , 431    ,
     432    , 434    , 435    , 439    , 440    ,1023    , 442    ,
     443    , 444    , 445    , 446    , 447    , 448    , 449    ,
     450    , 451    , 452    , 453    , 454    , 455    , 456    /
      END
      BLOCK DATA DIR10A
      COMMON/FIXDIR/NAME(900),LOC(900),NCOUNT
      DATA   (NAME(K),K=253,378)/
     6HIA87Y ,6HIA90X ,6HIA90Y ,6HIA90Z ,6HIA91X ,6HIA91Y ,6HIA91Z ,
     6HIA92X ,6HIA92Y ,6HIA92Z ,6HIA93X ,6HIA93Y ,6HIA93Z ,6HIA94X ,
     6HIA94Y ,6HIA94Z ,6HIA95X ,6HIA95Y ,6HIA95Z ,6HIDISPL,6HIECONT,
     6HIMTHD ,6HINDACC,6HINDACM,6HINDADD,6HINDADT,6HINDAED,6HINDAEP,
     6HINDAER,6HINDAIM,6HINDAPC,6HINDATH,6HINDATM,6HINDAUT,6HINDA01,
     6HINDA02,6HINDA03,6HINDA04,6HINDA05,6HINDA06,6HINDA07,6HINDA08,
     6HINDA09,6HINDA10,6HINDA11,6HINDA12,6HINDA13,6HINDA14,6HINDA15,
     6HINDA16,6HINDA17,6HINDA18,6HINDA19,6HINDA20,6HINDA21,6HINDA22,
     6HINDA23,6HINDA24,6HINDA25,6HINDA26,6HINDA27,6HINDA28,6HINDA29,
     6HINDA30,6HINDA31,6HINDA32,6HINDA33,6HINDA34,6HINDA35,6HINDA36,
     6HINDA37,6HINDA38,6HINDA39,6HINDA40,6HINDA41,6HINDA42,6HINDA43,
     6HINDA44,6HINDA45,6HINDA46,6HINDA47,6HINDA48,6HINDA49,6HINDA50,
     6HINDA51,6HINDA52,6HINDA53,6HINDA54,6HINDA55,6HINDA56,6HINDA57,
```

```
     *   6HINDA58,6HINDA59,6HINDA60,6HINDA61,6HINDA62,6HINDA63,6HINDA64,
     *   6HINDA65,6HINDA66,6HINDA67,6HINDA68,6HINDA69,6HINDA70,6HINDA71,
     *   6HINDA72,6HINDA73,6HINDA74,6HINDA75,6HINDA76,6HINDA77,6HINDA78,
     *   6HINDA80,6HINDA81,6HINDA82,6HINDA83,6HINDA84,6HINDA85,6HINDA86,
     *   6HINDA87,6HINDA90,6HINDA91,6HINDA92,6HINDA93,6HINDA94,6HINDA95/
      DATA   (NAME(L),L=379,504)/
     *   6HINDBAD,6HINDCCT,6HINDCGT,6HINDCNV,6HINDDCR,6HDTMIN ,6HINDDYR,
     *   6HINDEDT,6HINDFLT,6HINDFPA,6HINDFPD,6HINDFPR,6HINDGCR,6HINDGFP,
     *   6HINDGRT,6HINDGSR,6HINDGVT,6HINDHET,6HINDHTL,6HINDIDT,6HINDINP,
     *   6HINDINV,6HINDJUP,6HINDLAT,6HINDLOF,6HINDLEA,6HINDMAC,6HINDPLA,
     *   6HINDPIP,6HINDPSC,6HINDRMC,6HINDRTT,6HINDCJ1,6HINDCJ2,6HINDCJ3,
     *   6HINDCJ4,6HINDRJ5,6HINDRJ6,6HINDRJ7,6HINDRJ8,6HINDRJ9,6HINDR10,
     *   6HINDR11,6HINDR12,6HINDR13,6HINDR14,6HINDR15,6HINDR16,6HINDR17,
     *   6HINDR18,6HINDR19,6HINDR2Y,6HINDR20,6HINDR21,6HINDR22,6HINDR23,
     *   6HINDR24,6HINDR25,6HINDR26,6HINDR27,6HINDR28,6HINDSCL,6HINDSPL,
     *   6HINDSQT,6HINDSTF,6HINDSTR,6HINDSTY,6HINDSWT,6HINDTFF,6HINDTSO,
     *   6HINDVPC,6HINDWGT,6HINDWIN,6HINDXYS,6HINDXZS,6HSTGTOL,6HIPCONT,
     *   6HIP1   ,6HIP2   ,6HIQ   ,6HITCASE,6HIT10W ,6HIT10X ,6HIT10Y ,
     *   6HIT10Z ,6HIT11W ,6HIT11X ,6HIT11Y ,6HIT11Z ,6HIVARBH,6HNCASE ,
     *   6HNSTAGE,6HOG01  ,6HOMGPR ,6HOMGRT ,6HOMXGR ,6HOMZGR ,6HPA77P ,
     *   6HPE77R ,6HPHGZD ,6HPHIAD ,6HPHIBD ,6HPHIGD ,6HPHILD ,6HPHIRD ,
     *   6HPHITD ,6HPHLZD ,6HPI77R ,6HPI77R1,6HPM77R ,6HPRNC  ,6HPSIUD ,
     *   6HPSIPD ,6HPST7D ,6HQA77D ,6HQE77R ,6HQI77R ,6HQI77R1,6HQM77R ,
     *   6HRA77D ,6HRJ77N ,6HREM   ,6HRE77F ,6HRL77R ,6HRG77N ,6HRHN7F /
      DATA   (LOC(K),K=253,378)/
     *      457    ,  458    ,  459    ,  460    ,  461    ,  462    ,  463    ,
     *      464    ,  465    ,  466    ,  467    ,  468    ,  469    ,  470    ,
     *      471    ,  472    ,  473    ,  474    ,  475    ,  365    ,  364    ,
     *        1    , 1042    ,  477    ,  478    , 1038    ,  930    ,  479    ,
     *      480    ,  481    ,  482    ,  484    ,  485    ,  486    ,  487    ,
     *      488    ,  489    ,  490    ,  491    ,  492    ,  493    ,  494    ,
     *      495    ,  496    ,  497    ,  498    ,  499    ,  500    ,  501    ,
     *      502    ,  503    ,  504    ,  505    ,  506    ,  507    ,  508    ,
     *      509    ,  510    ,  511    ,  512    ,  513    ,  514    ,  515    ,
     *      516    ,  517    ,  518    ,  519    ,  520    ,  521    ,  522    ,
     *      523    ,  524    ,  525    ,  526    ,  527    ,  528    ,  529    ,
     *      530    ,  531    ,  532    ,  533    ,  534    ,  535    ,  536    ,
     *      537    ,  538    ,  539    ,  540    ,  541    ,  542    ,  543    ,
     *      544    ,  545    ,  546    ,  547    ,  548    ,  549    ,  550    ,
     *      551    ,  552    ,  553    ,  554    ,  555    ,  556    ,  557    ,
     *      558    ,  559    ,  560    ,  561    ,  562    ,  563    ,  564    ,
     *      565    ,  566    ,  567    ,  568    ,  569    ,  570    ,  571    ,
     *      572    ,  573    ,  574    ,  575    ,  576    ,  577    ,  578    /
      DATA   (LOC(L),L=379,504)/
     *      579    , 1036    , 1037    , 1040    ,  580    ,  581    ,  582    ,
     *      583    ,  587    ,  588    ,  589    ,  590    ,  591    ,  592    ,
     *      593    ,  594    ,  595    ,  596    ,  597    ,  598    ,  600    ,
     *      601    ,  602    ,  603    ,  604    ,  605    , 1041    ,  607    ,
     *      608    ,  609    ,  612    , 1134    ,  615    ,  616    ,  617    ,
     *      618    ,  619    ,  620    ,  621    ,  622    ,  623    ,  624    ,
     *      625    ,  626    ,  627    ,  628    ,  629    ,  630    ,  631    ,
     *      632    ,  633    , 1039    ,  634    ,  635    ,  636    ,  637    ,
     *      638    ,  639    ,  640    ,  641    ,  642    ,  643    ,  646    ,
     *     1035    ,  648    ,  652    ,  653    ,  654    ,  655    ,  656    ,
     *      657    ,  658    ,  659    ,  660    ,  661    ,    4    ,  662    ,
     *        6    ,    7    ,    8    ,  663    ,  911    ,  912    ,  913    ,
     *      914    ,  915    ,  916    ,  917    ,  918    ,    5    ,  665    ,
     *      667    ,  669    ,  671    ,  673    ,  675    ,  676    ,  678    ,
     *      679    ,  680    ,  682    ,  684    ,  685    ,  689    ,  695    ,
     *      699    ,  705    , 1170    , 1172    ,  710    ,  713    ,  714    ,
     *      719    ,  723    ,  724    ,  726    , 1178    , 1180    ,  729    ,
     *      732    , 1182    ,  735    ,  765    ,  766    ,  771    ,  773    /
      END
      BLOCK DATA DIRDA
      COMMON/FIXDIR/NAME(9,0),LOC(900),NCOUNT
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DATA NCOUNT/894/
      DATA   (NAME(M),M=505,637)/
     *    6HRHOAS  ,6HRHOSP  ,6HRI77R  ,6HRI77R1,6HRM77R  ,6HRNCR   ,6HRNGMN  ,
     *    6HRNKHG  ,6HRNKHL  ,6HRNFHG  ,6HRNRHL  ,6HRNTP   ,6HRNTR   ,6HRNTY   ,
     *    6HRNYHG  ,6HRNYHL  ,6HRPHLF  ,6HRPLZF  ,6HRP77F  ,6HR777F  ,6HSGRCB  ,
     *    6HSIDEP  ,6HSIGDD  ,6HSIGN   ,6HSIGZD  ,6HSIG7D  ,6HSIG7R1,6HSTEST  ,
     *    6HSTESTD,6HTA.1   ,6HTA.2   ,6HTA.3   ,6HTA.4   ,6HTA.5   ,6HTA.6   ,
     *    6HTA.7   ,6HTA10  ,6HTA11   ,6HTA14   ,6HTA77R  ,6HTE77P  ,6HTHL7D  ,
     *    6HTHL7J  ,6HTHTBD  ,6HTHTPJ  ,6HTHTRD  ,6HTHTTD  ,6HTIME   ,6HTIMES  ,
     *    6HTIME1  ,6HTIME3  ,6HTIME4  ,6HTIMSX  ,6HTIUPH  ,6HTMAX   ,6HTR77R  ,
     *    6HTSTGR  ,6HTSTGR1,6HTS77R  ,6HTS77R1,6HTVACP  ,6HTXA7P  ,6HTXB7P  ,
     *    6HTXE7P  ,6HTYA7P  ,6HTYB7P  ,6HTYE7P  ,6HTZA7P  ,6HTZB7P  ,6HTZE7P  ,
     *    6HT1     ,6HUM77F  ,6HUPSIF  ,6HUPSIF1,6HUPSIF2,6HU777F  ,6HU777F1,
     *    6HVA77F  ,6HVD77F  ,6HVGRVF  ,6HVG77F  ,6HVI77F  ,6HVM77F  ,6HVP77F  ,
     *    6HVS77F  ,6HVTHEF  ,6HV777F  ,6HV777F1,6HWM77F  ,6HWTR7P  ,6HW777F  ,
     *    6HW777F1,6HXCGBF  ,6HXCGRF  ,6HXD77D  ,6HXD77N  ,6HXE77F  ,6HXGW7F1,
     *    6HXGZ7F  ,6HXG77F  ,6HXI77F  ,6HXN77F  ,6HYA77P  ,6HYCGBF  ,6HYD77D  ,
     *    6HYD77N  ,6HYE77F  ,6HYGW7F1,6HYGZ7F  ,6HYG77F  ,6HYI77F  ,6HYN77F  ,
     *    6HZCGBF  ,6HZE77F  ,6HZGW7F1,6HZGZ7F  ,6HZG77F  ,6HZI77F  ,6HZN77F  ,
     *    6HZT.1   ,6HNDEBUG,6HCDEBUG,6HMPASS  ,6HNSMAIN,6HNS2NDY,6HINTOMP,
     *    6HOBTYPE ,6HNSTPUT,6HMASS   ,6HRX     ,6HRY     ,6HRZ     ,6HTHETAD/
      DATA   (LOC(M),M=505,637)/
     *      774    ,  775    ,1186    ,1188    ,  779    ,  782    ,1088    ,
     *      783    ,  784    ,  785    ,  785    ,  787    ,  788    ,  789    ,
     *      790    ,  791    ,  792    ,  793    ,  794    ,  795    ,  808    ,
     *      809    ,  812    ,  815    ,  586    ,  816    ,  818    ,  835    ,
     *      839    ,  850    ,  851    ,  852    ,  853    ,  854    ,  855    ,
     *      856    ,  857    ,  858    ,  861    ,  864    ,  865    ,  869    ,
     *     1190    ,  872    ,  875    ,  878    ,  881    ,  882    ,  885    ,
     *      886    ,  887    ,  888    ,  889    ,  890    ,  895    ,  897    ,
     *      898    ,  899    ,1200    ,1202    ,  900    ,  901    ,  902    ,
     *      903    ,  904    ,  905    ,  906    ,  907    ,  908    ,  909    ,
     *      910    ,  922    ,  927    ,  928    ,  929    ,1204    ,1206    ,
     *      933    ,1208    ,1212    ,  937    ,  938    ,  942    ,1216    ,
     *      945    ,1220    ,1224    ,1226    ,  952    ,  956    ,1228    ,
     *     1230    ,  960    ,  961    ,  962    ,  963    ,1232    ,  967    ,
     *      969    ,1238    ,1242    ,  975    ,  981    ,  983    ,  984    ,
     *      985    ,1246    ,  990    ,  992    ,1252    ,1256    ,  999    ,
     *     1006    ,1260    ,1009    ,1011    ,1266    ,1270    ,1017    ,
     *     1020    ,1272    ,1233    ,1314    ,1315    ,1316    ,1317    ,
     *     1318    ,1322    ,1323    ,1328    ,1333    ,1338    ,1343    /
      DATA   (NAME(N),N=638,759)/
     *    6HERDEG  ,6HRGR    ,6HNTIRES,6HPZERO  ,6HW      ,6HDELTAM,
     *    6HMOMENT,6HRF     ,6HVZ     ,6HIFD    ,6HPZERO  ,6HVZERO  ,6HA      ,
     *    6HP20    ,6HV20    ,6HA2     ,6HIL     ,6HS2T    ,6HES2    ,6HC2L    ,
     *    6HMASS2  ,6HMUS    ,6HCC     ,6HCE     ,6HC2C    ,6HC2E    ,6HNVGPT  ,
     *    6HNPP    ,6HMB     ,6HRLT    ,6HNDELTA,6HES     ,6HSB     ,6HSD21   ,
     *    6HSD22   ,6HSD23   ,6HSD24   ,6HSD25   ,6HSD11   ,6HSD12   ,6HSD13   ,
     *    6HSD14   ,6HSD15   ,6HS1     ,6HSS2    ,6HS3     ,6HS4     ,6HS5     ,
     *    6HS2D21  ,6HS2D22  ,6HS2D23  ,6HS2D24  ,6HS2D25  ,6HS2D11  ,6HS2D12  ,
     *    6HS2D13  ,6HS2D14  ,6HS2D15  ,6HS21    ,6HS22    ,6HS23    ,6HS24    ,
     *    6HS25    ,6HOMTD11,6HOMTD12,6HOMTD13,6HOMTD14,6HOMTD15,6HOMT1   ,
     *    6HOMT2   ,6HOMT3   ,6HOMT4   ,6HOMT5   ,6HAI     ,6HBI     ,6HDELTA1,
     *    6HDELTA2,6HDELTA3,6HDELTA4,6HDELTA5,6HDDELT1,6HDDELT2,6HDDELT3,
     *    6HDDELT4,6HDDELT5,6HISTAGE,6HPRTMIN,6HIPLT   ,6HISDF   ,6HISTPL1,
     *    6HISTPL2,6HISTPL3,6HISTPL4,6HISTPL5,6HIN     ,6HZN     ,6HYN     ,
     *    6HN      ,6HICS    ,6HCDCH   ,6HSSH    ,6HXCH    ,6HYCH    ,6HZCH    ,
     *    6HIB     ,6HITO    ,6HHF     ,6HNF     ,6HXRF    ,6HHRF    ,6HDELTAH,
     *    6HKP     ,6HTI     ,6HNLRI   ,6HVS     ,6HXS     ,6HTS     ,6HVATO   ,
     *    6HALPHTO,6HHS     ,6HALPDES,6HIAP    ,6HVD     ,6HDELVE  ,6HEPSGS  ,
     *    6HALPHUS,6HALPHDL,6HHCG    ,6HDELEPS,6HRFH    ,6HPGS    ,6HDELSIG/
      DATA   (LOC(N2),N2=638,769)/
     *     1348    ,1349    ,1350    ,1355    ,1360    ,1365    ,
     *     1370    ,1375    ,1380    ,1381    ,1382    ,1387    ,1392    ,
     *     1397    ,1402    ,1407    ,1412    ,1413    ,1418    ,1423    ,
```

```
     *     1428    ,1433    ,1438    ,1443    ,1448    ,1453    ,1458     ,
     *     1459    ,1464    ,1465    ,1466    ,1467    ,1472    ,1477     ,
     *     1479    ,1481    ,1483    ,1485    ,1487    ,1489    ,1491     ,
     *     1493    ,1495    ,1497    ,1499    ,1501    ,1503    ,1505     ,
     *     1507    ,1509    ,1511    ,1513    ,1515    ,1517    ,1519     ,
     *     1521    ,1523    ,1525    ,1527    ,1529    ,1531    ,1533     ,
     *     1535    ,1537    ,1539    ,1541    ,1543    ,1545    ,1547     ,
     *     1549    ,1551    ,1553    ,1555    ,1557    ,1562    ,1567     ,
     *     1568    ,1569    ,1570    ,1571    ,1572    ,1573    ,1574     ,
     *     1575    ,1576    ,1577    ,1578    ,1579    ,1580    ,1581     ,
     *     1582    ,1583    ,1584    ,1585    ,1586    ,1587    ,1592     ,
     *     1597    ,1602    ,1603    ,1604    ,1605    ,1606    ,1607     ,
     *     1608    ,1613    ,1614    ,1615    ,1616    ,1617    ,1618     ,
     *     1619    ,1620    ,1621    ,1622    ,1623    ,1624    ,1625     ,
     *     1626    ,1627    ,1628    ,1629    ,1630    ,1631    ,1632     ,
     *     1633    ,1634    ,1635    ,1636    ,1637    ,1638    ,1639     /
      DATA   (NAME(N3),N3=770,874)/
     *     6HREY   ,6HPHIC  ,6HTTTD  ,6HXTD   ,6HHTD   ,6HVXTD  ,6HVHTD   ,
     *     6HIIR   ,6HLD    ,6HJA    ,6HTL    ,6HTU    ,6HKE    ,6HPH     ,
     *     6HTSP   ,6HTRV   ,6HTCH   ,6HTBK   ,6HISS   ,6HILP   ,6HIAS    ,
     *     6HIC    ,6HXRF1  ,6HIT1   ,6HXRF2  ,6HIT2   ,6HH1    ,6HIH1    ,
     *     6HH2    ,6HIH2   ,6HHP1   ,6HIHR1  ,6HHP2   ,6HIHR2  ,6HTR1    ,
     *     6HITR1  ,6HTR2   ,6HITR2  ,6HTBK1  ,6HIBK1  ,6HTBK2  ,6HIBK2   ,
     *     6HDELQN ,6HRFALPH,6HDELALA,6HPSH   ,6HTST   ,6HDELQF ,6HDELPD1 ,
     *     6HDELQTU,6HDELQL ,6HDELQU ,6HINDLG ,6HPFB   ,6HDELBA ,6HPSR    ,
     *     6HRFPSI ,6HDPSIA ,6HPSPSI ,6HDELRL ,6HDELRU ,6HRFPHI ,6HDPHIA  ,
     *     6HPSA   ,6HDELPL ,6HDELPU ,6HTF    ,6HNDF   ,6HIR    ,6HNB     ,
     *     6HNLR   ,6HNTU   ,6HK2    ,6HK3    ,6HK4    ,6HMBC   ,6HPD     ,
     *     6HDELTAW,6HCMLCD1,6HMBL   ,6HMBU   ,6HDELHS ,6HDELRRD,6HDELA   ,
     *     6HNED1  ,6HHR    ,6HDELQDE,6HMANLOG,6HRFALP2,6HPSH2  ,6HDELOC  ,
     *     6HDELQC2,6HALPDL ,6HAUXICP,6HPITCHP,6HYAWAUP,6HROLLAP,6HBRAKAP ,
     *     6HCONTRP,6HTHROAP,6HINDICP,6HRWHGR ,6HHRD1  ,6HXR    ,6HXRD1    /
      DATA   (LOC(N4),N4=770,874)/
     *     1640    ,1641    ,1642    ,1643    ,1644    ,1645    ,1646     ,
     *     1647    ,1648    ,1649    ,1650    ,1651    ,1652    ,1657     ,
     *     1658    ,1659    ,1660    ,1661    ,1662    ,1663    ,1664     ,
     *     1665    ,1670    ,1671    ,1676    ,1677    ,1682    ,1683     ,
     *     1688    ,1699    ,1694    ,1695    ,1700    ,1701    ,1706     ,
     *     1707    ,1712    ,1713    ,1718    ,1719    ,1724    ,1725     ,
     *     1730    ,1731    ,1732    ,1733    ,1734    ,1735    ,1736     ,
     *     1737    ,1738    ,1739    ,1740    ,1741    ,1742    ,1743     ,
     *     1744    ,1745    ,1746    ,1747    ,1748    ,1749    ,1750     ,
     *     1751    ,1752    ,1753    ,1754    ,1759    ,1764    ,1769     ,
     *     1774    ,1779    ,1794    ,1785    ,1790    ,1801    ,1806     ,
     *     1807    ,1808    ,1813    ,1818    ,1823    ,1824    ,1825     ,
     *     1826    ,1827    ,1828    ,1829    ,1830    ,1831    ,1832     ,
     *     1833    ,1834    ,1835    ,1836    ,1837    ,1838    ,1839     ,
     *     1840    ,1841    ,1842    ,1843    ,1844    ,1845    ,1846     /
      DATA NAME(875)/4HCASK/
      DATA LOC(875)/1848/
      END
      BLOCK DATA DIRBDA
      COMMON/FIXDIR/NAME(900),LOC(900 ,NCOUNT
      DATA (NAME(K3),K3=875,894)/
     *   6HINDFLX,6HNMODE ,6HGMASS1,6HGFREQ ,6HSXMOD ,6HSYMOD ,6HSZMOD ,
     *   6HSKC    ,6HTXMOD ,6HDCMODE,6HARMODE,6HNPTS  ,6HOUTMOD,6HROIS  ,
     *   6HPF     ,6HGQ    ,6HGQD1  ,6HGQD2  ,6HIFLX   /
      DATA (LOC(K4),K4=876,894)/
     *   1892    ,1893    ,1894    ,1914    ,1934    ,2034    ,2134    ,
     *   2234    ,2259    ,2319    ,2379    ,2499    ,2500    ,3700    ,
     *   3760    ,3880    ,3900    ,3920    ,3940    /
      END
      OVERLAY(TOLA,2,0)
      PROGRAM TOLAN2
      COMMON/CONTRO/ICONTR,CONTR1,ICONT2,ICONT3
```

```
      IF(ICONTR-2)1,2,2
      CALL AUTS
      GO TO 19
2     CALL AUTSPR
19    CONTINUE
      END
      SUBROUTINE AUTS
C
C          A U X I L I A R Y   C O M P U T A T I O N S
C
C        INPUT REQUIRED (FROM DIRCOM)
C     XG77F, YG77F, ZG77F - LOCAL GEOCENTRIC DISPLACEMENTS
C     XG77F1, YG77F1, ZG77F1, AL2, AL27S1, AN2,AN277S,AM2, AM27S1
C     RGR
C
      COMMON/DIRCOM/DM1(73),TRR,DM20(42),ALPHD,DM2(3),ALPHR1,DM3(15),
     *AMASS,DM4(25),ARCFF,DM5(11),DELTAD,DM6(3),RCTAR1,DM7(147),
     *DELPD,DM8,DELQD,DM9,DELRD,DM11(61),FX07P,DM12(23),
     *GREFF,DM13(17),HGC7F,DM14(211),INDSTE,DM10(47),PHIPD,
     *DM15(23),PSIPD,DM35(74),RHOAS,DM38(31),
     *SGAMA,DM16( 75),TIME,DM17(50),VA77F,
     *DM18(3),VG77F,DM37(23),XGW7F1,DM19(22),
     *YGW7F1,DM21(18),ZGW7F1,DM36(120),AL2,DM65,
     *AL27S1,DM32(13),AM2,DM69,AM27S1,DM33(9),
     *AN2,DM66,AN277S,DM34(43),
     *U777F,DM24(33),XG77F,DM63,XG77F1,DM61(11),
     *YG77F,DM62,YG77F1,DM63(11),ZG77F,DM64,ZG77F1,DM25(53)
      COMMON/DIRCOM/NSTRUT,DM26(26),RGR,NTIRES(5),
     *RZERO(5),DM27(10),MOMENT(5),DM28(85),MB(5),DM29(82),
     *OMT1(2),OMT2(2),OMT3(2),OMT4(2),OMT5(12),
     *DELTA1,DELTA2,DELTA3,DELTA4,DELTA5,DM30(14)
      COMMON/DIRCOM/
     1IN,ZN(5),YN(5),N(5),ICS,CDCH,SSH,XCH,YCH,ZCH,
     2IJ(5),ITO,HF,NF,XRF,HRF,DELTAH,RP,TI,NLRI,VS,XS,TS,
     3VATO,ALPHTO,HS,ALPDES,IAP,VD,DELVE,EPSGS,ALPHDS,ALPHDL,
     4HCG,DELEPS,RFH,PGS,DELSIG,RFY,PHIC,TTD,XTD,HTD,VXTD,
     5VHTD,IIR,LD,CA,TL,TU,KE(5),PM,TSP,TRV,TCH,TBK,ISS,
     6ILR,IBJ,IC(5),XRF1,IT1(5),XRF2,IT2(5),H1,IH1(5),
     7H2,IH2(5),HR1,IHR1(5),HR2,IHR2(5),TR1,ITR1(5),
     8TR2,ITR2(5),TOK1,IOK1(5),TOK2,ICK2(5),DELQN,
     9RFALPH,DELACA,PSH,TST,DELOF,DELFD1,DELQTO,DELQL,DELQU,INDLG
      COMMON/DIRCOM/
     1RFB,DELBA,PSF,RFPSI,DPSIA,PSPSI,DELRL,DELRU,RFPHI,DPHIA,
     2PSA,DELPL,DELPU,TF(5),NOF(5),IR(5),ND(5),NLR(5),NTO(5),K2,
     3K3(5),K4(11),MBC(5),PD,DELTAW,OMECD1,DM67(4),MBL(5),MBU(5),
     4DELHS,DELRRD,DELA,NED1,HR,DELQDC,DM68,RFALP2,PSH2,
     *DELQC,DELQC2,ALPDL,DM31(9),XR,XRD1,HRD1,DELQI,DM39(44),
     *DDM40(2068)
C
C        INPUT REQUIRED FROM LGEAR
C     RG11, RG13, RG31, RG33
C
      COMMON/LGAUTS/RG11,RG13,RG31,RG33,MA(5),VAXLE(5)
C
C        OUTPUT GENERATED
C     XR, YR, ZR   - C.G. POSITION RELATIVE TO RUNWAY COORDINATE SYSTEM
C     XRD1, YRD1, ZRD1, - VELOCITY IN RUNWAY COORDINATE SYSTEM
C     HR, HRD1 - ALTITUDE AND ALT. RATE IN RUNWAY COORDINATE SYSTEM
C     PSIPD1 - YAW RATE IN EARTH AXES FOR P-Y-R SEQUENCE
C     PHIPD1 - ROW RATE IN EARTH AXES FOR P-Y-R SEQUENCE
C
      COMMON/AUTSC/TR,TC,TDX,
     1TOA,TOB,TD(5),IRA,IRB,IRC,ICA,ICB,
     2KA,KB,KEA,KEB,KT,NA,NNB,NC,NO(5),NBA,NBB,NBC,
     3NLRA,NLRB,NLRC,NDA,NDB,NDC,NTOA,NTOB,NTOC
```

```
C
      REAL KA,KB,NA,NNB,NC,ND,NBA,NBB,NBC,NLRA,
     1NLRB,NLRC,NDA,NDB,NDC,NTOA,NTOB,NTOC,
     2MBC,MDL,MBU,MG,NTIRES,MOMENT,MA,LR,N,NE01,
     3N01,ISS
C
      COMMON/AUTREC/YR,ZR,YRD1,ZRD1,PSIPD1,PHIPD1,
     1VE,VAD,GAMPRL,QR,CLP,COR,LR,UR,HGS,RR,HEA,HE,
     2HET,HPA,HPI,PHIDES,TX,TH,XRFD1,HRFD1,AXR,AHR,
     3ALPHAE,ALDD1,ALPHET,DELRN,BETAST,DELRD2,OM,
     4PHIE,PHIET,DELPD2,DELQD1,DELRD1,DELPD1,
     5ND1(5),OMEGTR(5),OMTRD1(5),OMEG E(5),PSIE,PSIET
      COMMON/EXEAUT/ADD1M1,SWT1,TIME1,ALPDD1,PNDES(5)
      COMMON/AUTSAC/ALPOR1,QDESR,CLRR,ALPO,CORR,DELRNN
C
C     POSITION AND VELOCITY RELATIVE TO RUNWAY COORDINATE SYSTEM
C
      LOGICAL SWT1,SWT2
      DIMENSION DELTA(5),OMET(2,5)
      EQUIVALENCE (DELTA1,DELTA(1)),(OMT1(1),OMET(1,1))
      DATA RADDEG,DEGRAD/57.2957795,.01745329/
      IPR=1
      GO TO 2
      ENTRY AUTSPR
      IPR=2
 2    DELTS=TIME-TIME1
      TIME1=TIME
      XR=RG11*(XG77F-RGR)+RG13*ZG77F
      YR=YG77F
      ZR=RG31*(XG77F-RGR)+RG33*ZG77F
      XRD1=RG11*XG77F1+RG13*ZG77F1
      YRD1=YG77F1
      ZRD1=RG31*XG77F1+RG33*ZG77F1
      HR=-ZR
      HRD1=-ZRD1
C
C     EULER ANGLE DER. FOR PITCH, YAW, ROLL SEQUENCE
C
      PSIPD1=0.
      IF((1.-AL2*AL2).LT.0.)GO TO 1
      PSIPD1=AL27S1/SQRT(1.-AL2*AL2)
 1    PHIPD1=(AN2*AM27S1/AM2-AN277S)/(AM2*(1.+(AN2/AM2)**2))
      IF(IPR.EQ.2)CALL AUTPR1
C
C          L O G I C   T O   D E T E R M I N E   P R O B L E M   P H A S E
C
C        INPUT REQUIRED(FROM DIRCOM)
C     ITO    = 0, FOR LANDING
C     ITO    NOT = 0, FOR TAKE OFF
C     HF     FLARE ALTITUDE
C     NF     INDICATOR TO STOP AT END OF GLIDE SLOPE(NF=1)
C     XRF, HRF - INITIAL VALUES OF DESIRED TOUCHDOWN POINT
C        IN RUNWAY COORDINATE SYSTEM
C     DELTAH- ALTITUDE ABOVE HRF FOR HOLD MODE
C     KP     - IMPACT INDICATOR
C     TI     - IMPACT TIME SET. IF KP=0, NO TI NEEDED
C     NLRI   - INDICATOR TO STOP AT IMPACT
C            =1, STOP AT IMPACT
C     VS, XS, TS  - RUNWAY STOPPING CONDITIONS
C     VATO   - AIRSPEED FOR TAKEOFF
C     ALPHTO- ANGLE OF ATTACK FOR TAKEOFF
C     HS     - ALTITUDE ABOVE RUNWAY TO TERMINATE TAKEOFF
C
C        INPUT REQUIRED FROM LGEAR(DIRCOM)
C     DELTA1,DELTA2,DELTA3,DELTA4,DELTA5 - TIRE DEFLECTION FOR EACH GEAR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C          INPUT FROM SDF2(DIRCOM)
C       TIME, HGC7F, NSTRUT
C
        IF(ITO.NE.0)GO TO 20
        IF(HGC7F.GE.HF)GO TO 70
        IF(NF.EQ.1)GO TO 14
        IF((XR.GE.XRF).OR.(HR.LE.(HRF+DELTAH)))GO TO 13
C       CALL FLARE(ENTRY 1)
        CALL FLARE1(IPR)
C
C       CALL ENGINE FAILURE LOGIC
C
        CALL ENGFL
C       GO TO PITCH AUTOPILOT
        GO TO 90
 13     IF(KP.EQ.1)GO TO 10
        DO 11 I=1,NSTRUT
        IF(DELTA(I).GT.0.)GO TO 12
 11     CONTINUE
C       CALL FLARE(ENTRY 2)
        CALL FLARE2(IPR)
C
C       CALL ENGINE FAILURE LOGIC
C
        CALL ENGFL
C       GO TO PITCH AUTOPILOT
        GO TO 90
 12     KP=1
        TI=TIME
 10     IF(NLRI.EQ.1)GO TO 16
        IF((XRD1.LT.VS).OR.(XR.GT.XS).OR.
       C((TIME-TI).GT.TS))GO TO 16
C       GO TO LANDING ROLL
        GO TO 80
 14     WRITE(6,15)
 15     FORMAT(10X,18HEND OF GLIDE SLOPE)
        INOSTE=0
        RETURN
 16     WRITE(6,17)
 17     FORMAT(10X,21HAIRCRAFT HAS IMPACTED)
        INOSTE=0
        RETURN
C
C            G L I D E   S L O P E
C
C            INPUT REQUIRED FOR GLIDE SLOPE(FROM DIRCOM)
C       VD     - DESIRED INERTIAL VELOCITY DOWN THE GLIDE SLOPE
C       DELVE  - ALLOWED ERROR IN VE
C       EPSGS  - GLIDE SLOPE ELEVATION ANGLE
C       ALPHDS- LOWER LIMIT ON ALPHA
C       ALPHDL- UPPER LIMIT ON ALPHA
C       TTO    - DESIRED THRUST(INPUT ONLY IF IAP GREATER OR =3)
C       HCG    - DISTANCE BETWEEN WHEEL BOTTOM SURFACE AT C.G.
C       DELEPS- ALLOWED ANGULAR ERROR IN GLIDE SLOPE VERTICAL
C                POSITION FROM ORIGIN
C       RFH    - RATE FEED BACK CONTRIBUTION TO HET
C       PGS    - PHUGOID CONTROL SENSITIVITY
C       DELSIG- ALLOWED ANGULAR ERROR IN GLIDE SLOPE
C                HORIZONTAL POSITION FROM ORIGIN
C       RFY    - RATE FEED BACK CONTRIBUTION TO HPT
C       PHIC   - CONSTANT EULER ROLL ANGLE FOR CROSS RANGE CONTROL
C
C            INPUT FROM SDF2(DIRCOM)
C       VG77F, RHOAS, AMASS, GREFF, AREFF, XGW7F1, YGW7F1
```

```
C     ZGW7F1, HGC7F, XG77F, YG77F, ZG77F
C         INPUT FROM SOF2
C     XG77F1, YG77F1, ZG77F1
C
C         INPUT FROM LGEAR (DIRCOM)
C     RGP
C
C         OUTPUT GENERATED BY GLIDE SLOPE
C     VE     -GLIDE SLOPE INERTIAL VELOCITY ERROR
C     VAD    -DESIRED AIRSPEED IN GLIDE SLOPE
C     IAP    -PROBLEM PHASE INDICATOR
C     GAMMPR-FLIGHT PATH ANGLE FOR VAD
C     QR     -DYNAMIC PRESSURE AT VAD
C     CLR    -REQUIRED LIFT COEFFICIENT IN GLIDE SLOPE
C     CDR    -DRAG COEFFICIENT ASSOCIATED WITH CLR
C     LR     -LIFT AT CLR AND QR
C     DR     -DRAG AT CDR AND QR
C     THETAR -EULER PITH ANGLE FOR REQUIRED GLIDE SLOPE CONDITIONS
C     TTD    -DESIRED THRUST
C     HGS    -DESIRED GLIDE SLOPE ALTITUDE
C     RR     -DISTANCE TO GLIDE SLOPE ORIGIN
C     HEA    -GLIDE SLOPE VERTICAL POSITION ERROR ALLOWED
C     HE     -ACTUAL GLIDE SLOPE POSITION ERROR
C     HED1   -RELATIVE RATE LEAVING GLIDE SLOPE
C     HET    -TOTAL GLIDE SLOPE POSITION ERROR
C     HPT    -TOTAL GLIDE SLOPE HORIZONTAL POSITION ERROR
C     HPA    -GLIDE SLOPE HORIZONTAL POSITION ERROR ALLOWED
C     PHIDES-DESIRED EULER ROLL ANGLE
C
 70   IAP=1
      VE=VG77F-VD
      TMP1=EPSGS*DEGRAD
      IF(ABS(VE).LE.DELVE)GO TO 71
      TMP2=VD*COS(TMP1)-XGW7F1
      TMP3=VD*SIN(TMP1)-ZGW7F1
      VAD=SQRT(TMP2*TMP2+YGW7F1*YGW7F1+TMP3*TMP3)
      GAMMPR=ATAN(-TMP3/TMP2)
      GAMPRD=GAMMPR*RADDEG
      QR=.5*RHOAS*VAD*VAD
      CLR=AMASS*GREFF/(QR*AREFF)
      SWT2=.FALSE.
      QDESR=..
 76   DO 75 I=1,2
      ALPD=ALPDES
      CALL SAC31
      CLRR=CLR
      CALL SAC58
      ALPDES=ALPD
      CDR=CDRR
 75   CONTINUE
      LR=CLR*QR*AREFF
      DR=CDR*QR*AREFF
      THETAR=GAMMPR+ALPDES*DEGRAD
      TTD=(LR*SIN(GAMMPR)+DR*COS(GAMMPR))/COS(THETAR)
      IF(SWT2)GO TO 71
      CLR=(AMASS*GREFF+DR*SIN(GAMMPR)-TTD*SIN(THETAR))/
     C(COS(GAMMPR)*QR*AREFF)
      SWT2=.TRUE.
      GO TO 76
 71   HGS=ABS(XG77F-RGR)*TAN(TMP1)+HGG
      RR=SQRT((XG77F-RGR)**2+ZG77F*ZG77F)
      HEA=RR*DELEPS*DEGRAD
      HE=HGC7F-HGS
      HED1=-ZG77F1+XG77F1*TAN(TMP1)
      HET=HE+RFH*HED1
```

```
      IF(ABS(HET).GE.HEA)ALPDES=ALPDES-PGS*HET
      IF(ALPDES.LT.ALPHOS)ALPDES=ALPHOS
      IF(ALPDES.GT.ALPHOL)ALPDES=ALPHOL
      HPA=RR*DELSIG*DEGRAD
      HPT=YG77F+KFY*YG77F1
      IF(HPT.GT.HPA)GO TO 72
      IF(HPT.LT.(-HPA))GO TO 73
      PHIDES=0.
      GO TO 74
 72   PHIDES=-PHIC
      GO TO 74
 73   PHIDES=PHIC
C
C     CALL ENGINE FAILURE LOGIC
C
 74   CALL ENGFL
      IF(IPR.EQ.1)CALL AUTPR2
C     GO TO PITCH AUTOPILOT
      GO TO 90
C
C          T A K E   O F F   R O L L
C
C          INPUT REQUIRED(FROM DIRCOM)
C     VATO  -AIRSPEED FOR TAKEOFF
C     ALPHTO-ANGLE OF ATTACK FOR TAKEOFF
C     HS    -ALTITUDE ABOVE RUNWAY TO TERMINATE TAKEOFF
C
C         INPUT FROM SDF2(DIRCOM)
C     VA77F
C
C         OUTPUT GENERATED
C     ALPDES- DESIRED ANGLE OF ATTACK(INPUT WHEN IAP GREATER OR =3)
C     IAP   -PROBLEM PHASE INDICATOR
C
 20   IAP=5
      IF(VA77F.LT.VATO)GO TO 21
      IAP=6
      ALPDES=ALPHTO
 21   IF(HR.GT.HS)GO TO 22
C     CALL ENGINE FAILURE LOGIC
      CALL ENGFL
C     GO TO PITCH AUTOPILOT
      QDESR=0.
      GO TO 90
 22   WRITE(6,23)HR,HS
 23   FORMAT(1H0,5X,42HALTITITUDE ABOVE RUNWAY TO TERMINATE TAKEOFF/
     *6X,5HHR = E15.8,10X,5HHS = E15.8)
      INDSTE=0
      RETURN
C
C          L A N D I N G   R O L L
C
C          INPUT REQUIRED FOR LANDING ROLL(FROM DIRCOM)
C     TSP   - TIME AFTER IMPACT TO STAGE SPOILERS
C     TRV   - TIME AFTER IMPACT TO REVERSE SIGNAL
C     TCH   - TIME AFTER IMPACT TO CHUTE SIGNAL
C     TBK   - TIME AFTER IMPACT TO BRAKE SIGNAL
C     ISS = 0., SPOILER SIGNAL INITIALIZATION
C     ILR = 0, REVERSE SIGNAL INITIALIZATION
C     ICS = 0, CHUTE SIGNAL INITIALIZATION
C     IBS=0, BRAKE SIGNAL INITIALIZATION
C
C
C          INPUT FROM SDF2(DIRCOM)
C     TIME
```

```
C
C       OUTPUT GENERATED
C     ISS=1., ILR=1, ICS=1, IBS=1, IAP=4
C
 83   IAP=4
      TR=TIME-TI
      TRR=TR
      IF(TR.GE.TSP)ISS=1.
      IF(TR.GE.TRV)ILR=1
      IF(TR.GE.TCH)ICS=1
      IF(TR.GE.TBK)GO TO81
      IF(IPR.EQ..)CALL AUTPR4
      GO TO 18
 81   IBS=1
      IF(IPR.EQ..)CALL AUTPR4
C
C         B R A K E   F A I L U R E   L O G I C
C
C       INPUT REQUIRED FOR BRAKE FAILURE LOGIC(FROM DIRCOM)
C     TBK1  - FIRST TIME AFTER IMPACT TO STAGE BRAKE CONDITIONS
C     IBK1(I) ARRAY ASSOCIATED WITH TBK1
C     TBK2  - SECOND TIME AFTER IMPACT TO STAGE BRAKE CONDITIONS
C     IBK2(I) ARRAY ASSOCIATED WITH TBK2
C     NSTRUT- NO. OF STRUTS
C
C       OUTPUT GENERATED FOR LGEAR AND AUTS
C     IB(1) -
C
      IF(TR.LT.TBK1)GO TO 25
      DO 19 I=1,NSTRUT
 19   IB(I)=IBK1(I)
 25   IF(TR.LT.TBK2)GO TO 18
      DO 26 I=1,NSTRUT
 26   IB(I)=IBK2(I)
C
C     CALL ENGINE FAILURE LOGIC
C
 18   CALL ENGFL
C
C         P I T C H   A U T O P I L O T
C
C       INPUT REQUIRED FOR PITCH AUTOPILOT(PIRCOM)
C     DELON -NOMINAL HORIZONTAL STABILIZER TRIM POSITION
C     RFALPH-RATE FEED BACK FOR ALPHRE
C     DELALA-ERROR ALLOWED IN ALPHA CONTROL
C     PSH   -PILOT SENSITIVITY IN ALPHA OVER CONTROL
C     TST   -TIME AFTER IMPACT AT WHICH NOSE OVER BEGINS
C     DELOF -FINAL DELQD ON END OF NOSE OVER
C     DELFD1-HORIZONTAL STABLIZER RATE COMMAND IN ACHIEVING DELQF
C     DELQTO-HORIZONTAL STABLIZER POSITION FOR TAKE OFF
C     DELQL -LOWER LIMIT ON DELQDE
C     DELQU -UPPER LIMIT ON DELQDE
C     IAP   -PROBLEM PHASE INDICATOR
C
C       INPUT FROM SOF2(DIRCOM)
C     DELTS, ALPHR1, ALPHD
C
C       OUTPUT GENERATED BY PITCH AUTOPILOT
C     DELON -NOMINAL HORIZONTAL STABILIZER TRIM POSITION
C     ALPHAE-ANGLE OF ATTACK POSITION ERROR
C     ALFDD1-RATE OF CHANGE OF ALPDES
C     ALPHRE-ANGLE OF ATTACK RELATIVE RATE ERROR
C     ALPHET-TOTAL ANGLE OF ATTACK ERROR
C     DELQDE-DESIRED HORIZONTAL STABILIZER POSITION(DIRCOM)
C
```

```
      IF(IAP.EQ.4)GO TO 93
      IF(IAP.EQ.5)GO TO 91
      ALPO=ALPDES
      CALL SACS1
      IF(IAP.LT.2)GO TO 97
      RFALPH=RFALP2
      PSH=PSH2
      DELQC=DELQC2
97    ALPHAE=ALPHD-ALPDES
      IF(DELTS.LE.1.E-6)GO TO 96
      ALPDD1=(ALPDES-ADD1M1)/DELTS
      IF(ABS(ALPDD1).GT.ALPDL)ALPDD1=0.
96    ADD1M1=ALPDES
      ALDD1=ALPDD1
      ALPHRE=ALPHR1*RADDEG-ALPDD1
      ALPHET=ALPHAE+RFALPH*ALPHRE
      DELQDE=DELQN
      SIGNN=SIGN(1.,ALPHAE)
      IF(ABS(ALPHET).GE.DELALA)DELQDE=DELQN+PSH*ALPHET+SIGNN*DELQC
      GO TO 92
93    IF(TR.LE.TST)GO TO 95
      TMP1=DELFD1*(TR-TST)
      IF((DELQI-DELQF).GE.0.)GO TO 94
      DELQDE=DELQI+TMP1
      IF(DELQDE.GE.DELQF)DELQDE=DELQF
      GO TO 92
95    DELQI=DELQDE
      GO TO 92
94    DELQDE=DELQI-TMP1
      IF(DELQDE.LE.DELQF)DELQDE=DELQF
      GO TO 92
91    DELQDE=DELQTO
92    IF(DELQDE.LE.DELQL)DELQDE=DELQL
      IF(DELQDE.GE.DELQU)DELQDE=DELQU
      IF(IPR.EQ.0)CALL AUTPRS
C
C         Y A W   A U T O P I L O T
C
C         INPUT REQUIRED FOR YAW AUTOPILOT(DIRCOM)
C     RFB   -RATE FEED BACK FOR SIDESLIP
C     DELBA -SIDE SLIP ERROR ALLOWED
C     PSR   -PILOT SENSITIVITY IN SIDE SLIP OVER CONTROL
C     RFPSI -RATE FEED BACK FOR PSIE
C     DPSIA -EULER YAW ERROR ALLOWED
C     PSPSI -PILOT SENSITIVITY IN EULER YAW CONTROL
C     DELRL -LOWER RUDDER LIMIT ON DELRDE
C     DELRU -UPPER RUDDER LIMIT ON DELRDE
C     IAP   -PROBLEM PHASE INDICATOR
C
C         INPUT FROM SDF2(DIRCOM)
C     BETAD,BETAR1,PSIPD
C
C         OUTPUT GENERATED BY YAW AUTOPILOT
C     DELRN -NOMINAL RUDDER TRIM POSITION
C     BETAE -SIDESLIP POSITION ERROR
C     BETAET-TOTAL SIDE SLIP ERROR
C     DELRDE-DESIRED RUDDER POSITION
C     PSIE  -EULER YAW ANGLE POSITION ERROR
C     PSIET -TOTAL EULER YAW POSITION ERROR
C
      CALL SACS11
      DELRN=DELRNN
      IF(IAP.GT.2)GO TO 100
      BETAE=BETAD
      BETAET=BETAE+RFB*BETAR1*RADDEG
```

```
      DELRDE=DELRN
      IF(ABS(BETAET).GE.DELBA)DELRDE=DELRN+PSR*BETAET
      GO TO 101
  100 PSIE=PSIPD
      PSIET=PSIE+RFPSI*PSIPD1*RADDEG
      DELRDE=DELRN
      IF(ABS(PSIET).GE.DPSIA)DELRDE=DELRN+PSPSI*PSIET
  101 IF(DELRDE.LT.DELRL)DELRDE=DELRL
      IF(DELRDE.GT.DELRU)DELRDE=DELRU
      IF(IPR.EQ.1)CALL AUTPR6
C
C          R O L L   A U T O P I L O T
C
C          INPUT REQUIRED FOR ROLL AUTOPILOT(DIRCOM)
C     RFPHI -ROLL ANGLE RATE FEEDBACK
C     DPHIA -ROLL ANGLE ERROR ALLOWED
C     PSA   -PILOT SENSITIVETY IN ROLL OVER CONTROL
C     DELPL -LOWER AILERON LIMIT ON DELPDE
C     DELPU -UPPER AILERON LIMIT ON DELPDE
C     IAP   -PROBLEM PHASE INDICATOR
C
C          INPUT FROM SDP2(DIRCOM)
C     PHIPD
C
C          OUTPUT GENERADED BY ROLL AUTOPILOT
C     PHIE  -ROLL ANGLE ERROR
C     PHIET -TOTAL ROLL ANGLE ERROR
C     DELPDE-DESIRED AILERON POSITION
C
      IF(IAP.GT.3)GO TO 102
      PHIE=PHIPD-PHICES
      PHIET=PHIE+RFPHI*PHIPD1*RADDEG
      IF(ABS(PHIET).LT.DPHIA)GO TO 102
      DELPDE=PSA*PHIET
      GO TO 103
  102 DELPDE=0.
  103 IF(DELPDE.LT.DELPL)DELPDE=DELPL
      IF(DELPDE.GT.DELPU)DELPDE=DELPU
      IF(IPR.EQ.1)CALL AUTPR7
C
C     CALL THROTTLE AUTOPILOT
C
   33 CALL THAUTS(IPR)
C
C          B R A K E   A U T O P I L O T
C
C          INPUT REQUIRED (FROM DIRCOM)
C     MBC(I)- BRAKING CONSTANT FOR STRUT I
C     PD     - PER CENT SKID FOR CONTROLLED BRAKING
C     DELTAW- PER CENT SPEED ERROR ALLOWED
C     OMEGD1(I)-ACCELERATION TO OMEGTR(I) DESIRED
C     MBL(I)- LOWER BRAKING MOMENT ALLOWED FOR STRUT I
C     MBU(I)- UPPER BRAKING MOMENT ALLOWED FOR STRUT I
C     MB(I) - BREAKING MOMENT INITIALLY READ IN
C     NSTRUT- NUMBER OF INDEPENDENT STRUTS(LGEAR MOD)
C     RZERO(I)- RADIUS OF TIRES ON EACH STRUT(LGEAR MOD)
C     NTIRES(I) - NUMBER OF TIRES ON EACH STRUT(LGEAR MOD)
C     MOMENT(I) - MOMENT OF INERTIA OF SINGLE ROTATING
C                 WHEEL ON EACH STRUT(LGEAR MOD)
C     DELTA(I) - TIRE DEFLECTION FOR EACH GEAR(LGEAR MOD)
C
C          OUTPUT GENERATED
C     OMEGTR(I) - DESIRED WHEEL SPEED FOR PD FOR STRUT I
C     OMTRD1(I) - DESIRED WHEEL SPEED DECELLER FOR STRUT I
C     OMEGTE(I) - WHEEL SPEED ERROR FOR STRUT I
```

```
C     MB(I)     - BRAKING MOMENT FOR STRUT I
C
C         OUTPUT GENERATED FOR LGEAR (IN DIRCOM)
C     OMET(1,I),
C
C
C         INPUT REQUIRED (FROM LGEAR)
C     MA(I),OMET(1,I),VAXLE(I)
C
C          INPUT REQUIRED SDF2(DIRCOM)
C     AMASS, FXB7P
C
      DO 40 I=1,NSTRUT
      IF(IDS.NE.1)GO TO 41
      IF(IB(I))42,43,44
 42   OMET(1,I)=0.
      GO TO 41
 43   MB(I)=0.
      GO TO 41
 44   IF(IB(I).GT.1)GO TO 45
      TMP1=-(1.-PO)/(RZERO(I)-DELTA(I))
      OMEGTR(I)=VAXLE(I)*TMP1
      OMTRD1(I)=FXB7P*TMP1/AMASS
      OMEGTE(I)=OMET(1,I)-OMEGTR(I)
      TMP1=0.
      IF(ABS(OMET(1,I)).LE.1.E-10)GO TO 48
      TMP1=ABS(OMET(1,I))/OMET(1,I)
 48   IF(ABS(OMEGTE(I)).LT.ABS(DELTAW*OMEGTR(I)))GO TO 46
      IF(OMEGTE(I).GT.0.)GO TO 47
      MB(I)=TMP1*(MA(I)-OMEGD1*MOMENT(I)*NTIRES(I))
      GO TO 41
 45   MB(I)=MBC(I)
      GO TO 41
 46   MB(I)=TMP1*(MA(I)-OMTRD1(I)*MOMENT(I)*NTIRES(I))
      GO TO 41
 47   MB(I)=TMP1*(MA(I)+OMEGD1*MOMENT(I)*NTIRES(I))
 41   IF(MB(I).LT.MBL(I))MB(I)=MBL(I)
      IF(MB(I).GT.MBU(I))MB(I)=MBU(I)
 40   CONTINUE
      IF(IPR.EQ.1)CALL AUTPR9
C
C          C O N T R O L    R E S P O N S E
C
C         INPUT REQUIRED(FROM DIRCOM)
C     DELPD - CONTROL SURFACE DEFLECTION(INPUT AND OUTPUT)
C     DELQD - CONTROL SURFACE DEFLECTION(INPUT AND OUTPUT)
C     DELRD - CONTROL SURFACE DEFLECTION(INPUT AND OUTPUT)
C     DELHS - HORIZONTAL STABLIZER TIME RATE
C     DELRRD- RUDDER TIME RATE
C     DELA  - AILERON TIME RATE
C     NED1  - ENGINE TIME RATE
C     IN    - NO. OF ENGINES
C     DELTS - MAXIUM INTEGRATION INTERVAL
C     N(I)  - THROTTLE POSITION FOR EACH INGINE(INPUT AND OUTPUT)
C
      DELQD1=DELHS
      IF(DELQDE.LT.DELQD)DELQD1=-DELHS
      DELRD1=DELRRD
      IF(DELRDE.LT.DELRD)DELRD1=-DELRRD
      DELPD1=DELA
      IF(DELPDE.LT.DELPD)DELPD1=-DELA
      DO 50 I=1,IN
      IF(ND(I).EQ.0.)GO TO 51
      IF(SWT1)GO TO 52
      IF(ND(I))56,58,58


      IF(PMDES(I))52,59,59
```

```
      IF(PNDES(I))59,52,52
59    N(I)=-N(I)
52    IF(ND(I)-N(I))53,54,54
53    ND1(I)=-NED1
      GO TO 50
54    ND1(I)=NED1
      GO TO 50
51    N(I)=0.
      ND1(I)=0.
50    CONTINUE
      SWT1=.FALSE.
      DO 55 I=1,IN
55    PNDES(I)=ND(I)
      IF(ABS(DELQD1*DELTS).GE.ABS(DELQDE-DELQD))GO TO 60
      DELQD=DELQD+DELQD1*DELTS
      GO TO 61
60    DELQD=DELQDE
61    IF(ABS(DELRD1*DELTS).GE.ABS(DELRDE-DELRD))GO TO 62
      DELRD=DELRD+DELRD1*DELTS
      GO TO 63
62    DELRD=DELRDE
63    IF(ABS(DELPD1*DELTS).GE.ABS(DELPDE-DELPD))GO TO 64
      DELPD=DELPD+DELPD1*DELTS
      GO TO 65
64    DELPD=DELPDE
65    DO 67 I=1,IN
      IF(ABS(ND1(I)*DELTS).GE.ABS(ND(I)-N(I)))GO TO 66
      N(I)=N(I)+ND1(I)*DELTS
      GO TO 67
66    N(I)=ND(I)
67    CONTINUE
      IF(IPR.EQ.0)CALL AUTP10
      RETURN
      END
      SUBROUTINE FLARE1(IPR)
      EXTERNAL ASIN
C
C                F L A R E
C
C         INPUT REQUIRED(FROM DIRCOM)
C     XTD    - INITIAL SELECTED TOUCHDOWN CONDITIONS
C     HTD    - INITIAL SELECTED TOUCHDOWN CONDITIONS
C     VXTD   - INITIAL SELECTED TOUCHDOWN CONDITIONS
C     VHTD   - INITIAL SELECTED TOUCHDOWN CONDITIONS
C     IZR    - IF=0,INITIALLY SET XRF, HRF, XRFD1, HRFD1 TO XTD,
C              HTD, VXTD, VHTD
C     LD     - SUSPECTED LANDING DISTANCE
C     DA     - NUMBER OF INCREMENTS/DEG IN ALPHA SEARCH
C     TL     - LOWER THRUST LIMIT ALLOWED IN FLARE
C     TU     - UPPER THRUST LIMIT ALLOWED IN FLARE
C     PH     - MAX. TAIL DOWN INTERFERENCE ANGLE AT TOUCHDOWN
C     ALPHDS- LOWER LIMIT ON ALPHA
C     ALPHDL- UPPER LIMIT ON ALPHA
C     HR
C
C         INPUT REQUIRED(FROM LGEAR-DIRCOM)
C     RLT    - RUNWAY LENGTH(USED ALSO IN LGEAR)
C     ERDEG - RUNWAY ELEVATION ANGLE IN DEG(USED ALSO IN LGEAR)
C
C         INPUT REQUIRED(FROM SDF2-DIRCOM)
C     AMASS,DYNPP,AREFF,WTR7P
C     SGAMA,GAM7D,VG77F
C
      COMMON/DIRCOM/DM1(136),AMASS,DM2(25),AREFF,
     *DM3(191),DYNPP,DM19(58),GAM7D,DM4(391),
```

```
      *GAMA,DM13(13,),VG77F,DM18(18),ATR7P,DM5(365)
       COMMON/DIRCOM/DM5(26),ERDEG,DM7(116),RLT,DM8(126)
       COMMON/DIRCOM/DM9(30),XRF,HRF,DM15(10),ALPDES,IAP,DM16(3),
      *ALPHOS,ALPHOL,DM11(7),TTD,XTD,HTD,
      *VXTD,VHTD,IIR,LD,DA,TL,TU,DM21(5),PM,DM12(169),HR,DM14(16),
      *XR,XRD1,HRD1,DM17(45),DDM20(2068)
C
C         INPUT REQUIRED(FROM AUTS)
C      XR, XRD1, HRD1
C
C          OUTPUT GENERATED BY FLARE1
C      XRF    - ACTUAL SELECTED TOUCHDOWN CONDITIONS(DIRCOM)
C      HRF    - ACTUAL SELECTED TOUCHDOWN CONDITIONS(DIRCOM)
C      XRFD1  - ACTUAL SELECTED TOUCHDOWN CONDITIONS
C      HRFD1  - ACTUAL SELECTED TOUCHDOWN CONDITIONS
C      TX     - TIME TO TOUCHDOWN X CONSTRAINT
C      TH     - TIME TO TOUCHDOWN Y CONSTRAINT
C      DM     - INCREASE OF TOUCHDOWN POINT PAST XTD.
C      RSR    - TOUCHDOWN ALTITUDE RATE REQUIRED TO MEET TX=TH CONSTRAINT
C      AXR    - REQUIRED ACCELERATION TO MEET TOUCHDOWN CONDITIONS
C      AHR    - REQUIRED ACCELERATION TO MEET TOUCHDOWN CONDITIONS
C      GAMARD- AIRSPEED FLIGHT PATH ANGLE RELATIVE TO RUNWAY
C      TTD    - REQUIRED THRUST SATISFY AXR(DIRCOM)
C      IAP    - PROBLEM PHASE INDICATOR (DIRCOM)
C      ALPDES- DESIRED ANGLE OF ATTACK(DIRCOM)
C
       COMMON/AUTSC/TR,TC,TDX,
      1TDA,TDB,TD(5),IRA,IRB,IRC,ICA,ICB,
      2KA,KB,KEA,KEB,KT,NA,NNB,NC,ND(5),NBA,NBB,NBC,
      3NLPA,NLRB,NLRC,NDA,NDB,NDC,NTOA,NTOB,NTOC
C
       COMMON/AUTPRC/YR,ZR,YRD1,ZRD1,PSIPD1,PHIPD1,
      1VE,VAD,GAMPRD,QR,CLR,CDR,LR,DR,AGS,RR,HEA,HE,
      2HET,HPA,HPT,PHIDES,TX,TH,XRFD1,HRFD1,AXR,AHR,
      3ALPHAE,ALPDD1,ALPHET,DELPN,BETAET,DELRDE,DM,
      4PHIE,PHIET,DELPDE,DELQD1,DELRD1,DELPD1,
      5ND1(5),CMEGTR(5),OMTRD1(5),OMEGTE(5),PSIE,PSIET
C
       COMMON/AUTSAC/ALPDR1,QDESR,CL1,A1,CD1,DELRNN
       LOGICAL SWT1
       REAL LD,L1
       DATA RADDEG,DEGRAD/57.2957795,.01745329/
       IAP=2
       IF(IIR.EQ.1)GO TO 10
       IIR=1
       XRF=XTD
       HRF=HTD
       XRFD1=VXTD
       HRFD1=VHTD
10     TMP1=XRF-XR
       TMP2=XRD1+XRFD1
       TMP3=HRF-HR
       TMP4=HRFD1+HPD1
       TMP5=TMP3*TMP2/TMP4+XR-XTD
       TX=2.*TMP1/TMP2
       TH=2.*TMP3/TMP4
       IF(TX-TH)11,30,13
11     DM=TMP5
       IF((LD+DM).GT.RLT)GO TO 14
       XRF=XTD+DM
       TMP1=XRF-XR
       GO TO 30
14     XRF=XTD+RLT-LD
       TMP1=XRF-XR
       HRFD1=TMP2*TMP3/TMP1-HRD1
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      GO TO 30
13    DM=TMP5
      IF(DM.LT.0.)DM=0.
      XRF=XTD+DM
      TMP1=XRF-XR
      TX=2.*TMP1/TMP2
      IF(TX.LE.TH)GO TO 30
      RSR=2.*TMP3/TX-HRD1
      IF(RSR.GT.0.)GO TO 12
      HRFD1=RSR
      GO TO 30
12    HRFD1=0.
      XRFD1=HRD1*TMP1/TMP3-XRD1
30    TMP2=XRD1+XRFD1
      TMP4=HRFD1+HRD1
      AXR=(XRFD1-XRD1)*TMP2/(2.*TMP1)
      AHR=(HRFD1-HRD1)*TMP4/(2.*TMP3)
      GAMERR=(GAM7D-ERDEG)*DEGRAD
      TMP1=SIN(GAMERR)
      TMP2=COS(GAMERR)
      QDESR=ALPDR1+(AHR*TMP2-AXR*TMP1)/VG77F
      GAMMAD=ASIN(SGAMA)*RADDEG
      GAMAPD=GAMMAD-ERDEG
      GAMAPR=DEGRAD*GAMAPD
      TMP1=SIN(GAMAPR)
      TMP2=COS(GAMAPR)
      ERRAD=ERDEG*DEGRAD
      TMP3=WTR7P*SIN(ERRAD)
      TMP4=WTR7P*COS(ERRAD)
      TMP5=AMASS*AXR
      SWT1=.FALSE.
      A1=ALPHOS
C     CALL SACS TO FIND CL1 AND CD1 FOR ALPHA = A1
5     CALL SACS10
      CALL SACS9
      L1=CL1*DYNPP*AREFF
      D1=CD1*DYNPP*AREFF
      TMP6=DEGRAD*(A1+GAMAPD)
      TTH=(TMP5+L1*TMP1+D1*TMP2+TMP3)/COS(TMP6)
      IF(TTH.LT.TL)TTH=TL
      IF(TTH.GT.TU)TTH=TU
      IF(SWT1)GO TO 6
      TTD=TTH
      ALPDES=A1
6     AX=(TTH*COS(TMP6)-L1*TMP1-D1*TMP2-TMP3)/AMASS
      AH=(TTH*SIN(TMP6)+L1*TMP2-D1*TMP1-TMP4)/AMASS
      TAE=SQRT((AXR-AX)**2+(AHR-AH)**2)
      IF(SWT1)GO TO 7
      AE=TAE
      SWT1=.TRUE.
7     IF(TAE.GT.AE)GO TO 8
      AE=TAE
      TTD=TTH
      ALPDES=A1
8     A1=A1+DA
      IF(A1.LE.ALPHOL)GO TO 5
      PHIDES=0.
      IF(IPR.EQ.0)CALL AUTPR3
      RETURN
      ENTRY FLARE2
      GAMMAD=ASIN(SGAMA)*RADDEG
      GAMAPD=GAMMAD-ERDEG
      IAP=3
      PHIDES=0.
      IF((ALPDES+GAMAPD).GT.PM)ALPDES=PM-GAMAPD
```

```
      CALL STFL(2,2,PRT5)
      CALL STOVAR(2,TI,TR,DU,DU,DU,DU,DU,DU)
      CALL STFL(0,1,DU)
      RETURN
C
C     ENTRY FOR PRINTING PITCH AUTOPILOT DATA
C
      ENTRY AUTPR5
      IF(PITCHP.EQ.0)RETURN
      CALL LINES(1)
      WRITE(6,50)
 50   FORMAT(58X,15HPITCH AUTOPILOT)
      CALL STFL(2,6,PRT6)
      ALPHD1=ALPHR1*57.2957795
      CALL STOVAR(6,DELQN,ALPHAE,ALPHD1,ALPDD1,ALPHET,DELQDE,
     1DU,DU)
      CALL STFL(0,1,DU)
      RETURN
C
C     ENTRY FOR PRINTING YAW AUTOPILOT DATA
C
      ENTRY AUTPR6
      IF(YAWAUP.EQ.0)RETURN
      CALL LINES(1)
      WRITE(6,60)
 60   FORMAT(58X,13HYAW AUTOPILOT)
      CALL STFL(2,7,PRT7)
      BETAD1=BETAR1*57.2957795
      CALL STOVAR(7,DELRN,BETAD,BETAD1,BETAET,DELRDE,
     1PSIE,PSIET,DU)
      CALL STFL(0,1,DU)
      RETURN
C
C     ENTRY FOR PRINTING ROLL AUTOPILOT
C
      ENTRY AUTPR7
      IF(ROLLAP.EQ.0)RETURN
      CALL LINES(1)
      WRITE(6,70)
 70   FORMAT(58X,14HROLL AUTOPILOT)
      CALL STFL(2,3,PRT8)
      CALL STOVAR(3,PHIE,PHIET,DELPDE,DU,DU,DU,DU,DU)
      CALL STFL(0,1,DU)
      RETURN
C
C     ENTRY FOR PRINTING THROTTLE AUTOPILOT
C
      ENTRY AUTPR8
      IF(THROAP.EQ.0)RETURN
      CALL LINES(3)
      WRITE(6,80)
 80   FORMAT(58X,18HTHROTTLE AUTOPILOT)
      WRITE(6,81) (ND(I),I=1,IN)
 81   FORMAT(21X,9HND(IN) = E15.8,4(5X,E15.8))
      WRITE(6,82) (TD(I),I=1,IN)
 82   FORMAT(21X,9HTD(IN) = E15.8,4(5X,E15.8))
      RETURN
C
C     ENTRY FOR PRINTING BRAKE AUTOPILOT
C
      ENTRY AUTPR9
      IF(BRAKAP.EQ.0)RETURN
      CALL LINES(2)
      WRITE(6,90) (MB(I),I=1,NSTRUT)
      FORMAT(58X,15HBRAKE AUTOPILOT/21X,9HMB(I)  = E15.8,
```

PRECEDING PAGE BLANK NOT FILMED

85-86

```
     C4(5X,E15.8))
      IF(IDS.NE.1)RETURN
      CALL LINES(3)
      WRITE(6,91) (OMEGTR(I),I=1,NSTRUT)
 91   FORMAT(17X,13HOMEGATR(I) = E15.8,4(5X,E15.8))
      WRITE(6,92) (OMTR01(I),I=1,NSTRUT)
 92   FORMAT(15X,15HOMEGATR01(I) = E15.8,4(5X,E15.8))
      WRITE(6,93) (OMEGTE(I),I=1,NSTRUT)
 93   FORMAT(17X,13HOMEGATE(I) = E15.8,4(5X,E15.8))
      RETURN
C
C     ENTRY FOR PRINTING CONTROL RESPONSE DATA
C
      ENTRY AUTPLO
      IF(CONTRP.EQ.0)GO TO 12
      CALL LINES(1)
      WRITE(6,100)
 100  FORMAT(58X,16HCONTROL RESPONSE)
      CALL STFL(2,6,PRT9)
      CALL STOVAR(6,DELQD1,DELRD1,DELPD1,DELQD,DELRD,
     1DELPD,DU,DU)
      CALL STFL(2,1,DU)
      CALL LINES(2)
      WRITE(6,101) (N01(I),I=1,IN)
 101  FORMAT(24X,6HN01 = E15.8,4(5X,E15.8))
      WRITE(6,102) (N(I),I=1,IN)
 102  FORMAT(24X,6HN    = E15.8,4(5X,E15.8))
C
C     WRITING OF INDICATORS
C
 12   IF(INDICP.EQ.0)RETURN
      CALL LINES(2)
      WRITE(6,11) IAP,(IC(I),I=1,4),(IB(J),J=1,5)
 11   FORMAT(61X,10HINDICATORS/30X,5HIAP =I2,10X,
     C7HIC(I) =I2,1H,,I2,1H,,I2,1H,,I2,10X,
     C7HIB(I) =I2,1H,,I2,1H,,I2,1H,,I2,1H,,I2)
      RETURN
      END
      SUBROUTINE THAUTS(IPR)
C
C     THROTTLE AUTOPILOT
C
C         INPUT REQUIRED (FROM DIPCOM)
C     TF(I) - FIX ARRAY FOR ENGINE I
C     NDF(I)- THROTTLE SETTING FOR TF(I)
C     IR(I) - REVERSE CAPABILITY ARRAY FOR ENGINE I
C     NB(I) - THROTTLE CONSTRAINT ARRAY FOR REVERSE FOR ENG. I
C     NLR(I)- REVERSE THROTTLE SETTING FOR REVERSE SIGNAL ON LANDING
C     NTO(I)- TAKE OFF THROTTLE SETTING FOR ENGINE I
C     KE(I) - KILL ENGINE OPTION FOR ENG. I FOR HOLD MODE
C       THE FOLLOWING ARE ENGINE LOAD CONSTANTS FOR VARIABLE
C          ENGINE COMBINATIONS
C     K2    - K(2)121
C     K3(I) - K(3)131, K(3)232, K(3)121, K(3)1231, K(3)1232
C     K4(I) - K(4)141, K(4)232, K(4)343, K(4)242, K(4)2342,
C             K(4)2343, K(4)1341, K(4)1343, K(4)12341,
C             K(4)12342, K(4)12343
C     IN    - NUMBER OF ENGINES
C     ILR   - REVERSE SIGNAL INITIALIZATION
C     IAP   - PROBLEM PHASE INDICATOR
C     TTD   - DESIRED THRUST
C
C          INPUT FROM SDF2(DIRCOM)
C     N(I)
C
```

```
      COMMON/DIRCOM/DM1(1585),IN,DM2(11),N(5),DM3(27),IAP,DM8(12),TTO,
     *DM4( 9),KE(5),DM5(6),ILR,DM9,IC(5),DM6(84),
     *TF(5),NDF(5),IR(5),NB(5),
     *NLR(5),NTO(5),K2,K3(5),K4(11),DM7(91),DDM8(2066)
C
C         INPUT FROM AUTS
C     TD(I), TDA, TDB, IC(I), NDA, NDB, NDC
C
C          OUTPUT GENERATED FOR AUTS
C     TD(I), TC, TDX, IRA, IRB, IRC, NA, NC, ND(I), NBA, NBB, NBC,
C     NLRA, NLRB, NLRC, NTOA, NTOB, NTOC, KA, KB, KT, KEA,
C     KEB, ICA, ICB, NNB
C
      COMMON/AUTSC/TR,TC,TDX,
     1TDA,TDB,TD(5),IRA,IRB,IRC,ICA,ICB,
     2KA,KB,KEA,KEB,KT,NA,NNB,NC,ND(5),NBA,NBB,NBC,
     3NLRA,NLRB,NLRC,NDA,NDB,NDC,NTOA,NTOB,NTOC
C
      REAL NDF,NB,N,NNB,NLR,NTO,K2,K3,K4,KA,KB,
     1NA,NC,ND,NBA,NBB,NBC,NLRA,NLRB,NLRC,
     2NTOA,NTOB,NTOC,NDA,NDB,NDC
C
      COMMON/AUTPRC/YR,ZR,YRD1,ZRD1,PSIPD1,PHIPD1,
     1VE,VAD,GAMPPD,QR,CLR,CDR,LR,DR,HCS,RR,HEA,HE,
     2HET,HPA,HPT,PHIDES,TX,TM,XRFD1,PRFD1,AXR,AHR,
     3ALPHAE,ALPDD1,ALPHET,DELRN,BETAET,DELRDE,DM,
     4PHIE,PHIET,DELPDE,DELDD1,DELRD1,DELPD1,
     5NJ1(5),OMEGTR(5),OMTRD1(5),OMEGTE(5),PSIE,PSIET
C
      INTEGER TF
C     CHECK ENGINE NO.
      DO 1 I=1,IN
      ND(I)=0.
    1 TD(I)=0.
      IF(IN-2)10,20,5
    5 IF(IN-4)30,40,40
C
C     SINGLE ENGINE LOGIC
C
   10 IF(TF(1).EQ.1)GO TO 11
      IF(IC(1).EQ.0)GO TO 12
C     SET UP DATA FOR COMMON ENGINE LOGIC
      TD(1)=TTD
      IRC=IR(1)
      NC=N(1)
      NBC=NB(1)
      NLRC=NLR(1)
      NTOC=NTO(1)
      TC=TD(1)
      KT=KE(1)
C     CALL COMMON ENGINE LOGIC
      CALL CENGL
      ND(1)=NDC
      GO TO 23
   11 ND(1)=NDF(1)
      IF(ILR.EQ.1)ND(1)=ENGREV(IR(1),N(1),NB(1),NLR(1))
      IF(IC(1).NE.0)GO TO 200
   12 ND(1)=0.
      GO TO 200
C
C     TWO ENGINE LOGIC
C
   20 IF((TF(1).EQ.1).OR.(TF(2).EQ.1))GO TO 21
      TDX=TTD
      ICA=IC(1)
```

```
      IRA=IR(1)
      NA=N(1)
      NBA=NB(1)
      NLRA=NLR(1)
      NTOA=NTO(1)
      KEA=KE(1)
      KA=K2
      ICB=IC(2)
      IRB=IR(2)
      NNB=N(2)
      NBB=NB(2)
      NLRB=NLR(2)
      NTOB=NTO(2)
      KEB=KE(2)
      KB=1.-K2
C     CALL COMMON TWO ENGINE LOGIC
      CALL CTENGL
      TD(1)=TDA
      TD(2)=TDB
      ND(1)=NDA
      ND(2)=NDB
      GO TO 200
 21   ND(1)=NDF(1)
      ND(2)=NDF(2)
      IF(ILR.EQ.1)ND(1)=ENGREV(IR(1),N(1),NB(1),NLR(1))
      IF(ILR.EQ.1)ND(2)=ENGREV(IR(2),N(2),NB(2),NLR(2))
      IF(IC(1).EQ.0)ND(1)=0.
      IF(IC(2).EQ.0)ND(2)=0.
      GO TO 200
C
C     THREE ENGINE LOGIC
C
C     NOTE: ONCE AN ENGINE IS STOPPED, IT STAYS FIXED
C           EXCEPT TO FAIL
C
 30   DO 31 I=1,3
      IF(TF(I).NE.1)GO TO 22
 31   CONTINUE
      DO 23 I=1,3
      ND(I)=NDF(I)
      IF(ILR.EQ.1)ND(I)=ENGREV(IR(I),N(I),NB(I),NLR(I))
      IF(IC(I).EQ.0)ND(I)=0.
 23   CONTINUE
      GO TO 200
 22   IF((TF(1).EQ.1).AND.(TF(3).EQ.1))GO TO 24
      IF(TF(2).NE.1)GO TO 50
      ND(2)=NDF(2)
      IF(ILR.EQ.1)ND(2)=ENGREV(IR(2),N(2),NB(2),NLR(2))
      IF(IC(2).EQ.0)ND(2)=0.
      CALL TFFS3(ND(2),TD(2))
 26   TDX=TTD-TD(2)
      I1=1
      I2=3
      I3=2
 27   ICA=IC(I1)
      ICB=IC(I2)
      IRA=IR(I1)
      IRB=IR(I2)
      NA=N(I1)
      NNB=N(I2)
      NBA=NB(I1)
      NBB=NB(I2)
      NLRA=NLR(I1)
      NLRB=NLR(I2)
      NTOA=NTO(I1)
```

```
      NTOB=NTO(I2)
      KEA=KE(I1)
      KEB=KE(I2)
      I4=I1+I3
      KA=K3(I4)
      KB=1.-K3(I4)
C     CALL COMMON TWO ENGINE LOGIC
      CALL CTENGL
      TO(I1)=TOA
      TO(I2)=TOB
      ND(I1)=NDA
      ND(I2)=NDB
      GO TO 200
 24   ND(1)=NDF(1)
      ND(3)=NDF(3)
      IF(ILR.EQ.1)ND(1)=ENGREV(IR(1),N(1),NB(1),NLR(1))
      IF(ILR.EQ.1)ND(3)=ENGREV(IR(3),N(3),NB(3),NLR(3))
      IF(IC(1).EQ.0)ND(1)=0.
      IF(IC(3).EQ.0)ND(3)=0.
      CALL TFFS9(NC(1),TO(1))
      CALL TFFS9(ND(3),TO(3))
      IF(IC(2).EQ.0)GO TO 51
      TO(2)=TTO-TO(1)-TO(3)
      I1=2
      I3=1
 28   IRC=IR(I1)
      NC=N(I1)
      NBC=NB(I1)
      NLRC=NLR(I1)
      NTOC=NTO(I1)
      TC=TO(I1)
      KT=KE(I1)
C     CALL COMMON ENGINE LOGIC
      CALL CENGL
      ND(I1)=NDC
      GO TO (29,56,57,29),I3
 29   GO TO 200
 51   ND(2)=0.
      GO TO 200
 50   IF(IC(1).EQ.0)GO TO 60
      IF(IC(2).NE.0)GO TO 55
      ND(2)=0.
      CALL TFFS9(ND(2),TO(2))
      GO TO 26
 60   ND(1)=0.
      CALL TFFS9(ND(1),TU(1))
      TOX=TTO-TO(1)
      I1=2
      I2=3
      I3=0
      GO TO 27
 55   IF(IC(3).EQ.0)GO TO 58
      TO(1)=TTO*K3(4)
      I1=1
      I3=2
      GO TO 28
 56   TO(2)=TTO*K3(5)
      I1=2
      I3=3
      GO TO 28
 57   TO(3)=TTO-TO(1)-TO(2)
      I1=3
      I3=4
      GO TO 28
      ND(3)=0.
```

```
      CALL TFFS9(ND(3),TD(3))
      I1=1
      I2=2
      I3=3
      GO TO 27
C
C     FOUR ENGINE LOGIC
C
40    DO 41 I=1,4
      IF(TF(I).NE.1)GO TO 42
41    CONTINUE
      DO 43 I=1,4
      ND(I)=NDF(I)
      IF(ILR.EQ.1)ND(I)=ENGREV(IR(I),N(I),NB(I),NLR(I))
      IF(IC(I).EQ.0)ND(I)=0.
43    CONTINUE
      GO TO 29
42    IF((TF(2).EQ.1).AND.(TF(3).EQ.1))GO TO 70
      IF((TF(1).NE.1).OR.(TF(4).NE.1))GO TO 80
      ND(1)=NDF(1)
      ND(4)=NDF(4)
      IF(ILR.EQ.1)ND(1)=ENGREV(IR(1),N(1),NB(1),NLR(1))
      IF(ILR.EQ.1)ND(4)=ENGREV(IR(4),N(4),NB(4),NLR(4))
      IF(IC(1).EQ.0)ND(1)=0.
      IF(IC(4).EQ.0)ND(4)=0.
      CALL TFFS9(ND(1),TD(1))
      CALL TFFS9(ND(4),TD(4))
      TDX=TTD-TD(1)-TD(4)
      I1=2
      I2=3
44    ICA=IC(I1)
      ICB=IC(I2)
      IRA=IR(I1)
      IRB=IR(I2)
      NA=N(I1)
      NNB=N(I2)
      NBA=NB(I1)
      NBB=NB(I2)
      NLRA=NLR(I1)
      NLRB=NLR(I2)
      NTOA=NTO(I1)
      NTOB=NTO(I2)
      KEA=KE(I1)
      KEB=KE(I2)
      KA=K4(I1)
      KB=1.-K4(I1)
      CALL CTENGL
      TD(I1)=TDA
      TD(I2)=TDB
      ND(I1)=NDA
      ND(I2)=NDB
      GO TO 27
70    ND(2)=NDF(2)
      ND(3)=NDF(3)
      IF(ILR.EQ.1)ND(2)=ENGREV(IR(2),N(2),NB(2),NLR(2))
      IF(ILR.EQ.1)ND(3)=ENGREV(IR(3),N(3),NB(3),NLR(3))
      IF(IC(2).EQ.0)ND(2)=0.
      IF(IC(3).EQ.0)ND(3)=0.
      CALL TFFS9(ND(2),TD(2))
      CALL TFFS9(ND(3),TD(3))
      TDX=TTD-TD(2)-TD(3)
      I1=1
      I2=4
      GO TO 44
      DO 81 I=1,4
```

```
      IF(IC(I).NE.0)GO TO 90
81    CONTINUE
      GO TO 95
90    DO 82 I=1,3
      IF(IC(I).NE.0)GO TO 91
82    CONTINUE
      GO TO 97
91    IF((IC(1).NE.0).OR.(IC(2).NE.0))GO TO 100
      IF(IC(4).NE.0)GO TO 100
      ND(1)=0.
      ND(2)=0.
      ND(4)=0.
      CALL TFFS9(0.,TD(1))
      TD(2)=TD(1)
      TD(4)=TD(1)
      TD(3)=TTD-TD(1)-TD(2)-TD(4)
      I1=3
      I2=1
83    IRC=IR(I1)
      NC=N(I1)
      NBC=NB(I1)
      NLRC=NLR(I1)
      NTOC=NTO(I1)
      TC=TD(I1)
      KT=KE(I1)
      CALL CENGL
      ND(I1)=NDC
      GO TO (84,101,102,103,111,121,131,141,151,152,161,162),I2
84    GO TO 200
95    DO 96 I=1,4
96    ND(I)=0.
      GO TO 200
97    DO 98 I=1,3
98    ND(I)=0.
      CALL TFFS9(0.,TD(1))
      TD(2)=TD(1)
      TD(3)=TD(1)
      TD(4)=TTD-TD(1)-TD(2)-TD(3)
      I1=4
      I2=1
      GO TO 83
100   IF((IC(1).EQ.0).AND.(IC(2).EQ.0))GO TO 110
      IF((IC(1).EQ.0).AND.(IC(3).EQ.0))GO TO 120
      IF((IC(1).EQ.0).AND.(IC(4).EQ.0))GO TO 130
      IF((IC(2).EQ.0).AND.(IC(3).EQ.0))GO TO 140
      IF(IC(1).EQ.0)GO TO 150
      IF(IC(2).EQ.0)GO TO 160
      I1=1
      I2=2
      TD(1)=TTD*K4(9)
      GO TO 83
101   TD(2)=TTD*K4(10)
      I1=2
      I2=3
      GO TO 83
102   TD(3)=TTD*K4(11)
      I1=3
      I2=4
      GO TO 83
103   TD(4)=TTD-TD(1)-TD(2)-TD(3)
      I1=4
      I2=1
      GO TO 83
110   ND(1)=0.
      ND(2)=0.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL TFFS9(0.,TD(1))
      TD(2)=TD(1)
      TD(3)=(TTD-TD(1)-TD(2))*K4(3)
      I1=3
      I2=5
      GO TO 83
111   TD(4)=(TTD-TD(1)-TD(2))*(1.-K4(3))
      I1=4
      I2=1
      GO TO 83
120   ND(1)=0.
      ND(3)=0.
      CALL TFFS9(0.,TD(1))
      TD(3)=TD(1)
      TD(2)=(TTD-TD(1)-TD(3))*K4(4)
      I1=2
      I2=6
      GO TO 83
121   TD(4)=(TTD-TD(1)-TD(3))*(1.-K4(4))
      I1=4
      I2=1
      GO TO 83
130   ND(1)=0.
      ND(4)=0.
      CALL TFFS9(0.,TD(1))
      TD(4)=TD(1)
      TD(2)=(TTD-TD(1)-TD(4))*K4(2)
      I1=2
      I2=7
      GO TO 83
131   TD(3)=(TTD-TD(1)-TD(4))*(1.-K4(2))
      I1=3
      I2=1
      GO TO 83
140   ND(2)=0.
      ND(3)=0.
      CALL TFFS9(0.,TD(2))
      TD(3)=TD(2)
      TD(1)=(TTD-TD(2)-TD(3))*K4(1)
      I1=1
      I2=8
      GO TO 83
141   TD(4)=(TTD-TD(2)-TD(3))*(1.-K4(1))
      I1=4
      I2=1
      GO TO 83
150   ND(1)=0.
      CALL TFFS9(0.,TD(1))
      TD(2)=(TTD-TD(1))*K4(5)
      I1=2
      I2=9
      GO TO 83
151   TD(3)=(TTD-TD(1))*K4(6)
      I1=3
      I2=10
      GO TO 83
152   TD(4)=(TTD-TD(1))*(1.-K4(5)-K4(6))
      I1=4
      I2=1
      GO TO 83
160   ND(2)=0.
      CALL TFFS9(0.,TD(2))
      TD(1)=(TTD-TD(2))*K4(7)
      I1=1
      I2=11
```

```
      GO TO 83
161   TD(3)=(TTD-TD(2))*K4(8)
      I1=3
      I2=12
      GO TO 83
162   TD(4)=(TTD-TD(2))*(1.-K4(7)-K4(8))
      I1=4
      I2=1
      GO TO 83
200   IF(IPP.EQ.1)CALL AUTPRA
      RETURN
      END
      FUNCTION ENGREV(IRC,NC,NBC,NLRC)
      REAL NC,NBC,NLRC
      IF(IRC.EQ.1)GO TO 1
2     ENGREV=1.
      RETURN
1     IF(NC.GT.NBC)GO TO 2
      ENGREV=NLRC
      RETURN
      END
      SUBROUTINE ENGFL
C
C         INPUT REQUIRED (FROM DIRCOM)
C     XRF1   - FIRST RUNWAY POSITION TO STAGE ENGINE FAILURE
C     IT1(I)- ARRAY ASSOCIATED WITH XRF1
C     XRF2   - SECOND RUNWAY POSITION TO STAGE ENGINE FAILURE
C     IT2(I)- ARRAY ASSOCIATED WITH XRF2
C     H1     - FIRST ALT. TO STAGE ENGINE FAILURE IN GLIDE SLOPE
C     IH1(I)- ARRAY ASSOCIATED WITH H1
C     H2     - SECOND ALT. TO STAGE ENGINE FAILURE IN GLIDE SLOPE
C     IH2(I)- ARRAY ASSOCIATED WITH H2
C     HR1    - FIRST ALT. TO STAGE ENGINE FAILURE IN FLARE
C     IHR1(I) ARRAY ASSOCIATED WITH HR1
C     HR2    - SECOND ALT. TO STAGE ENGINE FAILURE IN FLARE
C     IHR2(I) ARRAY ASSOCIATED WITH HR2
C     TR1    - FIRST TIME AFTER IMPACT TO STAGE ENGINE FAILURE
C     ITR1(I) ARRAY ASSOCIATED WITH TR1
C     TR2    - SECOND TIME AFTER IMPACT TO STAGE ENGINE FAILURE
C     ITR2(I) ARRAY ASSOCIATED WITH TR2
C     IAP    - PROBLEM PHASE INDICATOR
C
C         INPUT FROM SDF(DIRCOM)
C     HGC7F
C
      COMMON/DIRCOM/DM1(434),HGC7F,DM2(1193),IAP,DM3(35),
     *IC(5),XRF1,IT1(5),XRF2,IT2(5),H1,IH1(5),H2,IH2(5),
     *HR1,IHR1(5),HR2,IHR2(5),TR1,ITR1(5),TR2,ITR2(5),DM4(109),HR,
     *DM5(16),XR,DM6(47),DM7(2068)
C
C         INPUT FROM AUTS
C     XR, TR
C
C         OUTPUT GENERATED
C     IC(I) - INITIALIZATION OF FAILURE INDICATOR OF ENGINE I
C
      COMMON/AUTSC/TR,TC,TDX,
     1TDA,TDB,TD(5),IRA,IRB,IRC,ICA,ICB,
     2KA,KB,KEA,KEB,KT,NA,NNB,NC,ND(5),NBA,NBB,NBC,
     3NLRA,NLRB,NLRC,NDA,NDB,NDC,NTOA,NTOB,NTOC
C
C     ENGINE FAILURE LOGIC
C         (ONCE AN ENGINE FAILS-MUST REMAIN FAILED)
      IF(IAP-2)10,20,30
10    IF(HGC7F.GT.H1)GO TO 11
```

```
      DO 12 I=1,4
12    IC(I)=IH1(I)
11    IF(HGC7F.GT.H2)RETURN
      DO 13 I=1,4
13    IC(I)=IH2(I)
      RETURN
20    IF(HR.GT.HR1)GO TO 21
      DO 22 I=1,4
22    IC(I)=IHR1(I)
21    IF(HR.GT.HR2)RETURN
      DO 23 I=1,4
23    IC(I)=IHR2(I)
      RETURN
30    IF(IAP-4)20,40,50
40    IF(TR.LT.TR1)GO TO 41
      DO 42 I=1,4
42    IC(I)=ITR1(I)
41    IF(TR.LT.TR2)RETURN
      DO 43 I=1,4
43    IC(I)=ITR2(I)
      RETURN
50    IF(XR.LT.XRF1)GO TO 51
      DO 52 I=1,4
52    IC(I)=IT1(I)
51    IF(XR.LT.XRF2)RETURN
      DO 53 I=1,4
53    IC(I)=IT2(I)
      RETURN
      END
      SUBROUTINE CENGL
C
C     COMMON ENGINE LOGIC
C
C          INPUT FROM DIRCOM
C     ILR, IAP
C
      COMMON/DIRCOM/DM1(1628),IAP,DM2(33),ILR,DM3(228),DM4(2068)
C
C          INPUT FROM AUTS
C     TC, IRC, NC, NBC, KT, NTOC, NLRC
C
C          OUTPUT GENERATED FOR AUTS
C     NOC
C
      COMMON/AUTSC/TR,TC,TOX,
     1TOA,TOB,TO(5),IRA,IRB,IRC,ICA,ICB,
     2KA,KB,KEA,KEB,KT,NA,NNB,NC,NO(5),NBA,NBB,NBC,
     3NLRA,NLRB,NLRC,NOA,NOB,NOC,NTOA,NTOB,NTOC
C
      REAL N,NC,NBC,NLRC,NTOC,NOC
      IF(ILR.EQ.1)GO TO 10
      IF(IAP-3)11,12,13
11    CALL TFFS8(TC,N)
      IF(N.GT.0.)GO TO 20
      IF(IRC.NE.1)GO TO 21
      IF(NC.GT.NBC)GO TO 21
20    NOC=N
      RETURN
21    NOC=1.
      RETURN
12    IF(KT.NE.1)GO TO 11
      GO TO 21
13    IF(IAP.LT.5)GO TO 12
      NOC=NTOC
      RETURN
```

```
      IF(IRC.NE.1)GO TO 21
      IF(NC.GT.NBC)GO TO 21
      NDC=NLRC
      RETURN
      END
      SUBROUTINE CTENGL
C
C     COMMON TWO ENGINE LOGIC
C
C          INPUT FROM AUTS
C     ICA, ICB, IRB, NNB, NOB, NLRB, NTOB, TOB, KEB, NOC, TDX,
C     IRA, NA, NDA, NLRA, NTOA, TDA, KEA, KA, KB
C     TOB
C          OUTPUT GENERATED FOR AUTS
C     NDA, NDB, IRC, NC, NBC, NLRC, NTOC, TC, KT, NDB
C     TDA, TDB
C
      COMMON/AUTSC/TS,TC,TDX,
     1TDA,TDB,TD(5),IRA,IRB,IRC,ICA,ICB,
     2KA,KB,KEA,KEB,KT,NA,NNB,NC,ND(5),NBA,NBB,NBC,
     3NLRA,NLRB,NLRC,NDA,NDB,NDC,NTOA,NTOB,NTOC
C
      REAL KA,KB,NA,NNB,NC,NBA,NBB,NBC,NDA,NDB,NDC,NLRA,
     1NLRB,NLRC,NTOA,NTOB,NTOC
      IF((ICA.EQ.0).AND.(ICB.EQ.0))GO TO 30
      IF(ICA.EQ.0)GO TO 20
      IF(ICB.EQ.0)GO TO 10
      TDA=KA*TDX
      IRC=IRA
      NC=NA
      NBC=NBA
      NLRC=NLRA
      NTOC=NTOA
      TC=TDA
      KT=KEA
      CALL CENGL
      NDA=NDC
      TDB=KB*TDX
    5 IRC=IRB
      NC=NNB
      NBC=NBB
      NLRC=NLRB
      NTOC=NTOB
      TC=TDB
      KT=KEB
      CALL CENGL
      NDB=NDC
      RETURN
   10 NDB=0.
      CALL TFFS9(0.,TDB)
      TDA=TDX-TDB
      IRC=IRA
      NC=NA
      NBC=NBA
      NLRC=NLRA
      NTOC=NTOA
      TC=TDA
      KT=KEA
      CALL CENGL
      NDA=NDC
      RETURN
   20 NDA=0.
      CALL TFFS9(0.,TDA)
      TDB=TDX-TDA
      GO TO 5

   30 NDA=0.
      NDB=0.
      RETURN
      END
```

APPENDIX B

DERIVATION OF AERODYNAMIC WEIGHTING FACTORS

AERODYNAMIC WEIGHTING FACTORS

In the TOLA computer program, the aerodynamic effects are treated as concentrated loads in the form of total aerodynamic forces and moments acting at a reference point on the airframe. These quantities are calculated in the aerodynamics subprogram (SACS) and are defined as follows:

- a - axial force (body axis)
- y - side force (body axis)
- n_f - normal force (body axis)
- ℓ - moment about body X axis
- m - moment about body Y axis
- n - moment about body Z axis

To obtain realistic flexible body response to these aerodynamic loads, weighting effects or participation factors, PF(I), for the response of each normal mode are required.

An approach to determine the weighting effect for the Z translational degree of freedom is given below. A similar method can be used to determine the remaining participation factors.

In TOLA, the generalized aerodynamic force is calculated as the product of the total lift, the vertical modal deflection at the reference point, and a participation factor

$$Q_{A_Z} = n_f \, \phi_{Z_R} \, PF$$

If the actual generalized force due to aerodynamics is obtained by integrating the product of the pressure distribution and modal deflection over the lifting area, the expression for this quantity would be given as

$$Q_{A_Z} = \int_A \phi_Z \,(x,y)\; P(x,y)\; dA$$

Equating the two force expressions, an equation for the participation factor can be obtained.

$$PF = \frac{\int_A \phi_Z \,(x,y)\; P(x,y)\, dA}{n_f \, \phi_{Z_R}}$$

APPENDIX C

PROGRAM LISTING, PLTDAT COMPUTER PROGRAM

```
      PROGRAM PLTDAT(INPUT,TAPE3,TAPE10,OUTPUT,TAPE4)
      DIMENSION TITLE(15),BUF(400),NDIL(28),TBUF(400),
     *AVAL(4,6),
     *BMM(2,400),CMMDS(5),DEPVAR(5),LINE(7),NDVA(5)
      DATA CMMDS/4HPLOT,6HTITLE1,6HTITLE2,6HINDVAR,6HDEPVAR/
      DATA NPUT/5LINPUT/
      READ(NPUT,105)NPTS
105   FORMAT(I6)
      READ(3) NHL
      NW=0
      NDL=0
      NHL=NHL-1
      DO 10 N=1,NHL
      READ(3) NI,ND,(TITLE(I),I=1,NI)
106   FORMAT(2I6,3A6)
      IF (ND.GT.1) GO TO 7
      NS=NW+1
      NDL=NDL+1
      NDIL(NDL)=NI
100   FORMAT(13A10)
      ENCODE(130,100,TBUF(NS))(TITLE(I),I=1,NI)
      NW=NW+NI
      GO TO 10
7     DO 8 J=1,ND
      NS=NW+1
      NDL=NDL+1
      NDIL(NDL)=NI
      IF(J.LE.9)GO TO 111
      ENCODE(130,110,TBUF(NS))(J,TITLE(I),I=1,NI)
110   FORMAT(13(I2,A8))
      GO TO 112
111   ENCODE(130,113,TBUF(NS))(J,TITLE(I),I=1,NI)
113   FORMAT(13(*0*,I1,A8))
112   NW=NW+NI
8     CONTINUE
10    CONTINUE
      READ(3) IX
      DO 15 I=1,NW
      BMM(1,I)=1.E60
15    BMM(2,I)=-1.E60
      K=0
30    NDI=0
      DO 35 N=1,NDL
      NIL=NDIL(N)
      READ(3) (BUF(I+NDI),I=1,NIL)
      IF(EOF,3) 50,35
35    NDI=NDI+NIL
      READ(3) ENDPNT
      WRITE(4) (BUF(I),I=1,NW)
      DO 36 I=1,NW
      BMM(1,I)=AMIN1(BMM(1,I),BUF(I))
36    BMM(2,I)=AMAX1(BMM(2,I),BUF(I))
      K=K+1
      IF(K.GT.NPTS) GO TO 50
      GO TO 30
50    DO 51 I=1,15
51    TITLE(I)=10H
      NIV=0
      NDV=0
      KT1=0
      KT2=0
      CALL PLOTS(DUM,DUM,10)
      CALL PLOT(0.,1.,-3)
60    READ(NPUT,900)CMMD,LINE
900   FORMAT(10A10)
```

```
      IF(EOF,NPUT) 9.,59
59    DO 61 K=1,5
      IF(CMMD.EQ.CMMDS(K)) GO TO (80,62,63,65,66),K
61    CONTINUE
      PRINT 300,CMMD
300   FORMAT(* IMPROPER COMMAND-*,A10)
      STOP
62    ENCODE(40,100,TITLE(1))(LINE(I),I=1,4)
      KT1=KTB(TITLE(1),40)
      GO TO 60
63    ENCODE(40,100,TITLE(5))(LINE(I),I=1,4)
      KT2=KTB(TITLE(5),4.)
      GO TO 60
65    CALL FIND(TBUF,NW,LINE(1),NIV)
      IF(NIV.NE.0) GO TO 70
      PRINT 310,LINE(1)
310   FORMAT(* IMPROPER INDEPENDENT VARIABLE-*,A10)
      STOP
70    AVAL(1,1)=BMM(1,NIV)
      AVAL(2,1)=BMM(2,NIV)
      GO TO 60
66    NOV=0
      DO 69 N=1,5
      IF(LINE(N).EQ.1H ) GO TO 67
      CALL FIND(TBUF,NW,LINE(N),NOVA(N))
      IF(NOVA(N).NE.0) GO TO 68
      PRINT 320,LINE(N)
320   FORMAT(* IMPROPER DEPENDENT VARIABLE *,A10)
      STOP
67    NOV=N-1
      GO TO 60
68    NOX=NOVA(N)
      AVAL(1,N+1)=BMM(1,NOX)
      AVAL(2,N+1)=BMM(2,NOX)
      CALL SCALE(AVAL(1,N+1),6.,2,1)
69    CONTINUE
      NOV=5
      GO TO 60
80    IF(NIV.NE.0) GO TO 81
      PRINT 330
330   FORMAT(* NO INDEPENDENT VARIABLE*)
      STOP
81    IF(NOV.NE.0) GO TO 82
      PRINT 340
340   FORMAT(* NO DEPENDENT VARIABLES*)
      STOP
82    IF(KT1.NE.0) CALL LABEL(1.,6.5,10.,6.5,TITLE(1),KT1,
     *.3,2,0.,0.,0)
      IF(KT2.NE.0) CALL LABEL(1.,6.1,10.,6.1,TITLE(5),KT2,
     *.2,2,0.,0.,0)
      CALL PLOT(NOV*.6+.5,0.,-3)
      AXLEN=10.-NOV*.6
      CALL SCALE(AVAL(1,1),AXLEN,2,1)
      CALL AXIS(0.,0.,TBUF(NIV),-10,AXLEN,0.,AVAL(3,1),AVAL(4,1))
      DO 89 J=1,NOV
      JD=NOVA(J)
      REWIND 4
      IP=3
83    READ(4) (BUF(I),I=1,NW)
      IF(EOF,4) 85,84
84    VARIND=(BUF(NIV)-AVAL(3,1))/AVAL(4,1)
      VARDEP=(BUF(JD)-AVAL(3,J+1))/AVAL(4,J+1)
      IF(IP.EQ.3) CALL SYMBOL(VARIND,VARDEP,.1,J,0.,-1)
      IF(IP.NE.3) CALL PLOT(VARIND,VARDEP,2)
      IP=2
```

```
      GO TO 89
   85 NSUB=NOVA(J)
      CALL AXIS(-(J-1)*.6,0.,TBUF(NSUB),10,6.,90.,AVAL(3,J+1),
     *AVAL(4,J+1))
      CALL SYMBOL(-(J-1)*.6-.4,1.,.14,J,90.,-1)
   89 CONTINUE
      CALL PLOT(AXLEN+1.,0.,-3)
      GO TO 60
   90 CALL PLOT(0.,0.,999)
      STOP
      END
      FUNCTION KTB(ISTRG,IST)
      DIMENSION ISTRG(1)
      DO 20 I=1,IST
      KTB=IST+1-I
      NWD=(KTB+9)/10
      NBITS=6*MOD(KTB,10)
      ITST=ISHIFT(ISTRG(NWD),NBITS).AND.77B
      IF(ITST.NE.55B) RETURN
   20 CONTINUE
      KTB=0
      RETURN
      END
      SUBROUTINE FIND(TBUF,NW,COMP,N)
      DIMENSION TBUF(1)
      DO 10 N=1,NW
      IF(TBUF(N).EQ.COMP) RETURN
   10 CONTINUE
      N=0
      RETURN
      END
```

APPENDIX D

RIGID BODY EXAMPLE

```
STCASE TAB
ATAB.1        2
ATAB.2        2
ATAB880       2
ATAB10        2
ATAB11        2
ATAB12        2
ATAB15        2
ATAB16        2
ATAB51        2
ATAB52        2
ATAB53        2
ATAB56        2
ATAB57        2
CTAB.1        13
CTAB.2        13
FTAB.2        5
FTAB.3        20
TTAB10        50
VTAB.1        5
VTAB.2        5
VTAB.3        5
VTAB.4        5
VTAB.6        5
       TRA
REM    BCD 3SDF2-PLANPLANE
NCASE  BCD 1LROLL
REM    BCD 4INTEGRATION INFORMATION
IVARBH        0
TMAX          400.
AMINER        .00005
PRTMIN        0.
AMAXER        .001
DELTS         .001
PRINT         .05
REM    BCD 3REQD FOR SDF-2
AMASS         3874.899
HGC7F         1000.
GAM7D         -1.26.60
VG77F         296.2276
THTBD         10.3394
RWHGR         .0
INDAPC        1
INDAOD        1
INDFPA        1
INDPLA        1
INDACM        2
INDGRT        1
INDWGT        1
REM    BCD 5VEHICLE PHYSICAL PROP. DATA
INDVPC        1
XCGRF         0.
VTAB.1        2,0.,0.,5000.,1.
VTAB.2        2,0.,1.59702E+6,5000.,1.59702E+6
VTAB.3        2,0.,1.59702E+6,5000.,1.59702E+6
VTAB.4        2,0.,1.59702E+6,5000.,1.59702E+6
INDXZS        1
VTAB.6        2,0.,1.00E+5,5000.,1.00E+5
REM    BCD 4AERODYNAMIC INPUT DATA
INDAER        1
AREFF         1605.
D1RFF         37.7
D2RFF         56.7
INDA.1        1
ATAB.1        -7.9573427E-3,-5.5941985E-3
```

```
 TAB.2          1.4.2.979E-3,8.8931664E-4
INDA.2          1
INDA8.          1
ATAB8.          4.18951E-2,2..0844E-2
INDA10          1
ATAB10          -3.978321E-2,-3.9650349E-2
INDA11          1
ATAB11          5.7452448E-2,3.7645688E-2
INDA12          1
ATAB12          1.472.279E-3,1.2354312E-3
INDA15          1
ATAB15          -1.725E-2,-..45E-2
INDA16          1
ATAB16          1.25E-4,5.0E-5
INDA51          1
ATAB51          7.3516484E-3,1.7.27972E-3
INDA52          1
ATAB52          -5.2147652E-4,1..23892BE-3
INDA53          1
ATAB53          -3.1368631E-4,1.8.65268E-5
INDA56          1
ATAB56          8.7E-3,7.34.8182E-3
INDA57          1
ATAB57          2.0E-5,3.6363636E-5
REM        BCD  3ENGINE THRUST DATA
INDTFF          3
INDTSO          1
IT1.X           4
TTAB10          -2.,-1.5,-1.,-.5,0.,.5,1.,1.5,2.
TTAB10          0.,.1,.2,.3
TTAB10          0.,0.,0.,0.,0.
TTAB10          0.,0.,0.,0.
TTAB10          0.,0.,0.,0.,0.
TTAB10          0.,0.,0.,0.
TTAB10          0.,0.,0.,0.,0.
TTAB10          0.,0.,0.,0.
TTAB10          0.,0.,0.,0.,0.
TTAB10          0.,0.,0.,0.
IT1.W           9
REM        BCD  3LANDING GEAR DATA
NSTRUT          3
MASS            4.94...0.,27.19587,27.19587
RX              35..833,-2.83867,-2.83867
RY              0.,-8.3333,+8.3333
RZ              1.91657,.5.9583,.5.9583
THETAD          0.,0.,0.
ERDEG           0.
RGR             4.47
NTIRES          2.,3.,3.
RZERO           1..479167,1.13125,1.13125
W               .554.07,.604107,.604107
DELTAM          ..98333,.1958333,.19583333
RLT             1.5E+3
IFD             1
AI              1.32.4533E+5,3.8129286E+5,3.8129286E+5
BI              1.2.12257,1.412391,1.412391
FTAB.2          2,..,..,1.5E+4...
FTAB.3          6,..,..,..7,.336,.1,.336,.4,.336,1.,.336,1.0.,.336
MOMENT          1.2,2.6,2.6
MB              0.,0.,0.
RF              5.46333,6.8792,6.8792
VZ              0.
PZERO           3528...,4.32..,4.32..
VZERO           0.16.4583,.4716435,.4716435
F               .11.4166,.2673611,.2673611
```

10
14
19
23
28
32
37
41
46

```
-               1
INDOC.1         1
NSMAIN          2
CTAB.1          0.,.6667,1.,2.,3.
CTAB.1          136.701,136.701,443.92,443.92,443.92,443.92,443.92          6
CTAB.1          4443.92                                                    13
INDOC.2         1
CTAB.2          0.,1.6667,1.,2.,3.
CTAB.2          136.781,136.781,4.23E+4,4.23E+4,4.23E+4,4.23E+4             6
CTAB.2          4.23E+4,4.23E+4                                            12
MUS             .2,.2,.1
ES              .001333,.0016667,.0016657
SB              1.333,1.6667,1.6667
REM      BCD 4FLEXIBLE AIRFRAME DATA
INDFLX          0
REM      BCD 3AUTOPILOT DATA
REM      BCD 3A. ENGINE DATA
INDAUT          1
IN              2
ZN              -.55,-.55
YN              -14.1667,14.1667
N               0.,0.
REM      BCD 3B. DRAG CHUTE DATA
ICS             0
CDCH            .244
SSH             16.5.
XCH             -9.625
YCH             0.
ZCH             -2.41667
REM      BCD 5C. PHASE BEGIN-TERMINATE
ITO             0
HF              12.25
NF              0
XRF             0.
HRF             8.796
DELTAH          3.
NCRI            0
T1              0.
KP              0
VS              1..
XS              300..
TS              12.
HS              0.
REM      BCD 4G. HOLD MANEUVER DATA
ALPDES          11.5
TTD             0.
KE       INT    1,1
PM              11.5
REM      BCD 5H. LANDING ROLL MANEUVER DATA
TSP             0.
TRV             0.
TCH             7.
TBK             0.
ISS             0.
ICR             0
IBS             0
REM      BCD 5I. ENGINE FAILURE STAGE DATA
IC       INT    0,0
XRF1            0.
IT1      INT    0,0
XRF2            0.
IT2      INT    0,0
H1              0.
IH1      INT    0,0
H2              0.
```

```
T 2      INT 0,0
HR1          0.
IHR1     INT 0,0
HR2          0.
IHR2     INT 0,0
TR1          0.
ITR1     INT 0,0
TR2          0.
ITR2     INT 0
REM      BCD 5J. BRAKE COND.STAGE DATA
IB       INT 0,0,0
TBK1         1.
IBK1     INT 0,0,0
TBK2         1.
IBK2     INT 0,0,0
REM      BCD 4K. PITCH AUTOPILOT DATA
TST          .2
ALPDL        2.
RFALPH       0.
DELALA       .01
PSH          0.
PSH2         0.
RFALP2       0.
DELQF        8.
DELQTO       0.
DELQL        -10.
DELQU        +25.
DELFD1       5.25
REM      BCD 4L. YAW AUTOPILOT DATA
RFB          .5
DELBA        .05
PSR          10.
DPSIA        .05
RFPSI        .5
PSPSI        0.
DELRL        -20.
DELRU        +20.
REM      BCD 4M. ROLL AUTOPILOT DATA
RFPHI        .5
DPHIA        .05
PSA          -10.
DELPL        -12.
DELPU        +12.
REM      BCD 5N.THROTTLE AUTOPILOT DATA
TF       INT 0,0
NDF          0.,0.
IR       INT 0,0
NB           1.,1.
NLR          0.,0.
NTO          0.,0.
K2           1.
REM      BCD 4O. BRAKE AUTOPILOT DATA
MBC          0.,0.,0.
PD           .1
DELTAW       .001
OMECD1       20.
MBL          0.,0.,0.
MBU          0.,0.,0.
REM      BCD 5P. CONTROL RESPONSE DATA
DELHS        5.25
DELRRD       31.5
DELA         10.
NED1         .1
REM      BCD 4Q. INITIALIZATION
I*P          3
```

```
DELQD          4.56478
DELQDE         4.56478
DELQN          4.56478
DELPD          0.
DELRD          0.
MANLOG         1
PITCHP         1
REM       BCD 3DATA PLOT TAPE
IPLT           0
REM       BCD 2STAGING DATA
REM       BCD 4A.GEARS INTO PROGRAM
INDLG          0
ISTAGE         0
DECRES    BCD 1HR
STESTD         11.0
          TRA
INDLG          -1
REM       BCD 4B.SMOOTH IMPACT STAGE
AINCRS    BCD 3DDELT1DDELT2DDELT3
STEST          -.05,-.05,-.05
          TRA
PRINT          .02
DELTS          .005
AMAXER         .0005
PRTMIN         0.
AINCRS    BCD 1TIMES
STEST          .9
          TRA
ATABS1         7.3506484E-3,1.7027972E-3
ATABS2         -5.2147652E-4,1.0238928E-3
ATABS3         -3.1368631E-4,1.8065268E-5
ATABS6         0.,0.
ATABS7         0.,0.
REM       BCD 5C. EFFICIENT AMAXER STAGE
AINCRS    BCD 4DELTA1DELTA2DELTA3
STEST          .1,.1,.1
          TRA
PRINT          .05
AMAXER         .01
DELTS          .01
REM       BCD 4D.SMOOTH IMPACT STAGE
AINCRS    BCD 1DDELT1
STEST          -.05
          TRA
PRINT          .02
DELTS          .005
AMAXER         .0005
REM       BCD 5E. EFFICIENT AMAXER STAGE
AINCRS    BCD 1DELTA1
STEST          .1
          TRA
PRINT          .05
AMAXER         .01
DELTS          .01
INDSTF         1
          TRA
```

```
                    SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                        GENERALIZED COMPUTER PROGRAM

          INDSOF  2     CASE LROLL        STAGE  1     PAGE      8

     2SDF

 0.              0.              0.              0.              10.0000000E+00  29.0274042E+01  0.              59.0569126E+00
 0.              0.              0.              26.5337670E-02  29.6220760E+01  10.4252977E+01  29.6159990E+01  0.
59.9992281E-01  11.4999986E+00   0.              0.              0.             -11.6059853E-01  0.              10.3393988E+00
 0.              0.              57.7454851E-01  0.             -31.6515539E+00  12.4671000E+04  0.              0.
 0.              0.

     VPCS

38.7489900E+02

     TFFS

 0.              0.


                 0.              0.                              T(I)

     SAC1

29.7745819E-03-20.8749533E-03 60.9750569E-02  0.              0.              15.3281670E-03  0.

                                                                  FLARE
     AUTS

     ALPDES          PHIDES          TTO

11.5000000E+00  0.              0.

                                                          PITCH AUTOPILOT

     DELQN           ALPHAE          ALPHD1          ALPDD1          ALPHET          DELQDE

26.8544760E-01-14.0024531E-07  0.              0.             -14.0024531E-07 26.8544760E-01

INTEG RTN.      HT =  1.0000000E-03

     2SDF

 0.              0.              0.              0.              10.0000000E+00  29.0274042E+01  0.              59.0569126E+00
 0.              0.              0.              26.5337670E-02  29.6220760E+01  10.4252977E+01  29.6159990E+01  0.
59.9992281E-01  11.4999986E+00   0.              0.              0.             -11.6059853E-01  0.              10.3393988E+00
 0.              0.              90.1422933E-02  0.             -26.3302709E+00  12.4671000E+04  0.              0.
 0.              0.

     VPCS

38.7489900E+02
```

ORIGINAL PAGE IS
OF POOR QUALITY

SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
GENERALIZED COMPUTER PROGRAM

INDSOF 2 CASE LROLL STAGE 1 PAGE 9

TFFS

0. 0.

T(I)

0. 0.

SAC1

29.7745819E-03-20.8749533E-03 60.9750569E-02 0. 0. 15.3281670E-03 0.

FLARE

AUTS

ALPDES PHIDES TTD

11.5000000E+00 0. 0.

PITCH AUTOPILOT

DELON ALPHAE ALPHO1 ALPDO1 ALPHET DELODE

27.5607208E-01-14.0024531E-07 0. 0. -14.0024531E-07 27.5607208E-01

STAGE ON--DECR. HR

```
                        SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                              GENERALIZED COMPUTER PROGRAM

               INDSOF  2     CASE LROLL        STAGE  1      PAGE      10

          INDLG       -1
          REM    BCO  48.SMOOTH IMPACT STAGE
          AINCRS BCO  300ELT100ELT200ELT3
          STEST       -.05,-.05,-.05
                 TRA

    2SDF

0.              0.              0.              0.              10.0000000E+0. 29.0274042E+01 0.             59.0569120E+00
0.              0.              0.              26.5337670E-02 29.6220760E+01 10.4252977E+01 29.6159990E+01 0.
59.9992281E-01 11.4999986E+00  0.              0.              0.            -11.6059853E-01 0.             13.3393988E+00
0.              0.              90.1422983E-02 0.             -26.3302709E+00 12.4671000E+04 0.              0.
0.              0.

    LGEAR

    DELTA           P               P2              FT              SR             SF             AA              FC2

0.              35.2800000E+03  0.              0.              31.9713310E-01-38.9549765E+02-78.8562277E+01  0.
0.              40.3200000E+03  0.              0.              54.9319280E-01-10.7799996E+03-39.6383699E+01  0.
0.              40.3200000E+03  0.              0.              54.9319280E-01-10.7799996E+03-39.6383699E+01  0.

    MUVP            VGPT            FTRX            FTRY            FTRZ            MA              MB              DDELTA

33.6000000E-02 29.6159990E+01  0.              0.              -0.             0.              0.             -79.6895154E-01
33.6000000E-02 29.6159990E+01  0.              0.              -0.             0.              0.             -10.9046626E-01
33.6000000E-02 29.6159990E+01  0.              0.              -0.             0.              0.             -10.9046626E-01

    SD2             SD1             S               S2D2            S2D1            S2              OMETD1          OMET

0.              0.              0.              0.              0.              0.              0.              0.
0.              0.              0.              0.              0.              0.              0.              0.
0.              0.              0.              0.              0.              0.              0.              0.

    FTRA            FTRB            FTRC            MTX             MTY             MTZ             FXM             FYM

0.              0.              0.              0.              0.              0.              0.              0.

    FZM             LM              MM              NM

0.              0.              0.              0.

    VPCS

38.7489900E+02

    TFFS

0.              0.
```


ORIGINAL PAGE IS
OF POOR QUALITY

SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
GENERALIZED COMPUTER PROGRAM

```
                INOSDF  2      CASE LROLL        STAGE  1     PAGE      11

                                                           T(I)
         0.                   3.
   SAC1
29.7745819E-03-20.8749533E-03 64.9750569E-02  0.              0.            15.3281670E-03  0.

   AUTS                                                    FLARE
   ALPDES          PHIDES          TTD
11.5300000E+03  0.              0.

                                                    PITCH AUTOPILOT
   DELQN           ALPHAE          ALPHD1          ALPDD1          ALPHET          DELQDE
27.5339224E-01-14.0024531E-07  0.              0.             -14.0024531E-07 27.5339224E-01
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
```

SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
GENERALIZED COMPUTER PROGRAM

INDSOF 2 CASE LROLL STAGE 1 PAGE 12

```
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03
INTEG RTN.     HT =  1.0000000E-03

     2SDF

50.0000000E-03 50.0000000E-03 14.8031852E+00  0.                            96.9255670E-01 29.0027316E+01  0.                 59.3319920E+00
0.             27.2368959E-04  0.             26.5173101E-02 29.6034000E+01 10.4122497E+01 29.5967086E+01  0.
62.9389922E-01 11.5616902E+00  0.             12.6453882E-01  0.            -12.1824188E-01  0.                 10.3434483E+00
0.              0.             94.7824236E-02  0.                          -26.7931133E+00 12.4671000E+04  0.                  0.
0.              0.

     LGEAR

     DELTA          P              P2             FT             SR             SF             AA             FC2

0.             35.2800000E+03  0.              0.             31.6148848E-01-38.9549765E+02-78.8562277E+01  0.
0.             40.3200000E+03  0.              0.             49.9529874E-01-10.7799996E+03-39.6383699E+01  0.
0.             40.3200000E+03  0.              0.             49.9529874E-01-10.7799996E+03-39.6383699E+01  0.

     MUVP           VGPT           FTRX           FTRY           FTRZ           MA             MB             DDELTA

33.6000000E-02 29.5969707E+01  0.              0.            -0.              0.              0.            -76.6404145E-01
33.6000000E-02 29.5980272E+01  0.              0.            -0.              0.              0.            -78.2919333E-02
33.6000000E-02 29.5980272E+01  0.              0.            -0.              0.              0.            -78.2919333E-02

     SD2            SD1            S              S2D2           S2D1           S2             OMETD1         OMET

0.              0.             0.              0.             0.              0.              0.              0.
0.              0.             0.              0.             0.              0.              0.              0.
0.              0.             0.              0.             0.              0.              0.              0.

     FTRA           FTRB           FTRC           MTX            MTY            MTZ            FXM            FYM
```

```
                    SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                         GENERALIZED COMPUTER PROGRAM

              INDSOF  2      CASE LROLL        STAGE  1     PAGE     13
0.            0.            0.            0.            0.            0.            0.            0.
     FZM           LM            MM            NM
0.            0.            0.            0.
     VPCS
38.7489990E+02
     TFFS
0.            0.

                                                                  T(I)
              0.               0.
     SAC1
30.0142493E-03-21.9770289E-03 62.1246500E-02  0.            0.            12.2577406E-03  0.

     AUTS                                                          FLARE
     ALPDES         PHIDES         TTD
11.5000000E+00  0.             0.

                                                             PITCH AUTOPILOT
     DELQN          ALPHAE         ALPHD1         ALPDD1         ALPHET         DELQDE
28.3036950E-01 61.6902417E-03 12.6453882E-01  0.             61.6902417E-03 28.3036950E-01
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
INTEG RTN.    HT =   1.0000000E-03
```

SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
GENERALIZED COMPUTER PROGRAM

INDSDF 2 CASE LROLL STAGE 1 PAGE 14

```
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
INTEG RTN.      HT =  1.0000000E-03
```

2SDF

```
10.0000000E-02 10.0000000E-02 29.5966730E+00  0.                 93.7100927E-01 28.9775565E+01  0.                 59.6180378E+00
0.             48.1984450E-04  0.             26.5000399E-02 29.5844872E+01 10.3991476E+01 29.5772046E+01  0.
65.6393767E-01 11.6257361E+00  0.             12.9189599E-01  0.            -12.7133004E-01  0.                 13.3544058E+00
0.              0.             99.5793099E-02  0.               -27.2803123E+03 12.4671000E+04  0.                 0.
0.              0.
```

LGEAR

```
DELTA          P              P2             FT             SR             SF             AA             FC2

0.             35.2800000E+03  0.             0.             31.1847404E-01-38.9549765E+02-78.8562277E+01  0.
0.             40.3200000E+03  0.             0.             44.7098906E-01-10.7799996E+03-39.6393699E+01  0.
0.             40.3200000E+03  0.             0.             44.7098906E-01-10.7799996E+03-39.6393699E+01  0.
```

```
                         SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                               GENERALIZED COMPUTER PROGRAM
                 INDSDF  2     CASE LROLL       STAGE  1     PAGE     15
     MUVP            VGPT            FTRX            FTRY            FTRZ            MA              MB              DDELTA
33.6000000E-02 29.5776739E+01  0.              0.              -0.             0.              0.              -73.4934807E-01
33.6000000E-02 29.5834538E+01  0.              0.              -0.             0.              0.              -46.1091723E-02
33.6000000E-02 29.5809538E+01  0.              0.              -0.             0.              0.              -46.1091723E-02
     SD2             SD1             S               S2D2            S2D1            S2              OMETD1          OMET
0.              0.              0.              0.              0.              0.              0.              0.
0.              0.              0.              0.              0.              0.              0.              0.
0.              0.              0.              0.              0.              0.              0.              0.
     FTRA            FTRB            FTRC            MTX             MTY             MTZ             FXM             FYM
0.              0.              0.              0.              0.              0.              0.              0.
     FZM             LM              MM              NM
0.              0.              0.              0.
      VPCS
38.7489900E+02
      TFFS
0.              0.
                                                              T(I)
                0.              0.
      SAC1
30.2764715E-03-23.1185661E-03 63.3346129E-02  0.              0.              90.5470627E-04  0.
                                                              FLARE
     AUTS
     ALPDES          PHIDES          TTD
11.5000000E+00  0.              0.
                                                        PITCH AUTOPILOT
     DELON           ALPHAE          ALPHD1          ALPDD1          ALPHET          DELQDE
29.1245546E-01 12.5736066E-02 12.9189599E-01  0.              12.5736066E-02 29.1245546E-01
```

APPENDIX E

FLEXIBLE BODY EXAMPLE

```
STCASE TAB
ATAB01       2
ATAB02       2
ATAB80       2
ATAB10       2
ATAB11       2
ATAB12       2
ATAB15       2
ATAB16       2
ATAB51       2
ATAB52       2
ATAB53       2
ATAB56       2
ATAB57       2
CTAB.1       13
CTAB.2       13
FTAB.2       5
FTAB03       20
TTAB10       50
VTAB01       5
VTAB02       5
VTAB03       5
VTAB04       5
VTAB06       5
       TRA
REM    BCD 3SDF2-PLANPLANE
NCASE  BCD 1LROLL
REM    BCD 4INTEGRATION INFORMATION
IVARPH       0
TMAX         40.
AMINER       ..0005
PRTMIN       .0
AMAXER       .001
DELTS        .001
PRINT        .05
REM    BCD 3REQJ FOR SDF-2
AMASS        3674.899
HGC7F        10..
GAM7D        -1.16.60
VG77F        296.22076
THTBO        10.3394
RWHGR        .0
INDAPC       1
INDADD       1
INDFPA       1
INDPLA       1
INDACM       2
INDGRT       1
INDWGT       1
REM    BCD 5VEHICLE PHYSICAL PROP. DATA
INDVPC       1
XCGRF        .0
VTAB01       2,0.,0.,5000.,1.
VTAB02       2,0.,1.59702E+6,5000.,1.59702E+6
VTAB03       2,0.,1.59702E+6,5000.,1.59702E+6
VTAB04       2,0.,1.59702E+6,5000.,1.59702E+6
INDXZS       1
VTAB06       2,0.,1.0E+5,5000.,1.0E+5
REM    BCD 4AERODYNAMIC INPUT DATA
INDAER       1
AREFF        1605.
D1RFF        37.7
D2RFF        56.7
INDA01       1
ATAB01       -7.9573427E-3,-5.5941985E-3
```

```
ATAB12    1.4.2.979E-3,8.8931664E-4
INDA12    1
INDA10    1
ATAB10    4.18951E-2,2.01944E-2
INDA10    1
ATAB10    -3.978321E-2,-3.9650349E-2
INDA11    1
ATAB11    5.7452448E-2,3.7645688E-2
INDA12    1
ATAB12    1.472.279E-3,1.2354312E-3
INDA15    1
ATAB15    -1.725E-2,-1.45E-2
INDA16    1
ATAB16    1.25E-4,5.0E-5
INDA51    1
ATAB51    7.39164842-3,1.7027972E-3
INDA52    1
ATAB52    -5.2147852E-4,1.0238928E-3
INDA53    1
ATAB53    -3.1368631E-4,1.8065268E-5
INDA56    1
ATAB56    8.7E-3,7.3418182E-3
INDA57    1
ATAB57    2.0E-5,3.6363636E-5
REM   BCD 3ENGINE THRUST DATA
INDTFF    3
INDTSO    1
IT1SK     4
TTAB10    -2.,-1.5,-1.,-.5,0.,.5,1.,1.5,2.
TTAB10    0.,.1,.2,.3                                                10
TTAB10    0.,0.,0.,0.,0.                                             14
TTAB10    0.,0.,0.,0.                                                19
TTAB10    0.,0.,0.,0.,0.                                             23
TTAB10    0.,0.,0.,0.                                                28
TTAB10    0.,0.,0.,0.,0.                                             32
TTAB10    0.,0.,0.,0.                                                37
TTAB10    0.,0.,0.,0.,0.                                             41
TTAB10    0.,0.,0.,0.                                                46
IT10W     9
REM   BCD 3LANDING GEAR DATA
NSTRUT    3
MASS      4.9400000,27.19587,27.19587
RX        35.0833,-2.83867,-2.83867
RY        0.,-8.3333,+8.3333
RZ        1.91667,.539583,.539583
THETAD    0.,0.,0.
ERDEG     0.
RGR       4.47
NTIRES    2.,3.,3.
RZERO     1.3479167,1.13125,1.13125
W         .554167,.604167,.604167
DELTAM    .199333,.1958333,.19583333
RLT       1.5E+3
IFD       1
AI        1.32.4533E+5,3.8129286E+5,3.8129286E+5
BI        1.2112257,1.412391,1.412391
FTAB02    2,0.,0.,1.5E+4,0.
FTAB03    6,0.,0.,.007,.336,.1,.336,.4,.336,1.,.336,100.,.336
MOMENT    1.2,2.6,2.6
MB        0.,0.,0.
RF        5.48333,6.8792,6.8792
VZ        0.
PZERO     35280.,40320.,4.320.
VZERO     0.1614583,.4716435,.4716435
A         .1104166,.2673611,.2673611
```

```
IL          1
INDCL1      1
NSMAIN      2
CTAB.1      0.,1.6667,1.,2.,3.
CTABC1      136.781,136.781,443.92,443.92,443.92,443.92,443.92          6
CTABC1      443.92                                                        13
INDCC2      1
CTABC2      0.,1.6667,1.,2.,3.
CTABC2      136.781,136.781,4.23E+4,4.23E+4,4.23E+4,4.23E+4               6
CTABC2      4.23E+4,4.23E+4                                               12
MUS         .1,.1,.1
ES          .001333,.0016667,.0016667
SB          1.333,1.6667,1.6667
REM    BCD  4FLEXIBLE AIRFRAME DATA
INDFLX      1
NMODE       4
GMASS1      66.5617,73.4557,7..8341,13.1131
GFREQ       13.188,16.512,33.359,43.398
SZMOD       .08,-.10,-.10,-.02,-.0042,-.0042
SZMOD       -.153,.1393,.1393,-.065,.0112,.0112                       7
ARMODE      0.,0.,-.214,0.,-4.239E-4,0.
ARMODE      0.,0.,-.414,0.,3.543E-4,0.                                7
ARMODE      0.,0.,.1405,0.,-6.030E-4,0.                               13
ARMODE      0.,0.,.1644,0.,-3.034E-5,0.                               19
PF          0.,0.,1.944,0.,0.,0.
PF          0.,0.,-1.259,0.,0.,0.                                     7
PF          0.,0.,2.915,0.,0.,0.                                      13
PF          0.,0.,0.274,0.,0.,0.                                      19
NPTS        4
OUTMOD      0.,0.,0.,0.,0.,0.,0.,0.,.1970,-.0385,.0035,.0035
OUTMOD      0.,0.,0.,0.,0.,0.,0.,0.,.0800,-.0200,-.153,-.065          13
OUTMOD      0.,0.,0.,0.,0.,0.,0.,0.,-.100,-.0042,.1393,.0112          25
OUTMOD      0.,0.,0.,0.,0.,0.,0.,0.,.2330,-.2440,-.1072,-.125         37
ROIS        40.6667,0.,0.,35.0833,0.,1.91667
ROIS        -2.83367,3.3333,.509583,-32.3333,0.,0.                    7
REM    BCD  3PLOT TAPE DATA
IPLT        0
REM    BCD  3AUTOPILOT DATA
REM    BCD  3A. ENGINE DATA
INDAUT      1
IN          2
ZN          -.55,-.55
YN          -14.1667,14.1667
N           1.,1.
REM    BCD  3B. DRAG CHUTE DATA
ICS         0
CDCH        .244
SSH         1615.
XCH         -9.625
YCH         0.
ZCH         -2.41667
REM    BCD  5C. PHASE BEGIN-TERMINATE
ITO         0
HF          12.25
NF          0
XRF         0.
HRF         8.796
DELTAH      3.
NLRI        0
TI          0.
KP          0
VS          10.
XS          3000.
TS          12.
HS          0.
```

```
REM    BCD 4G. HOLD MANEUVER DATA
ALPDES     11.5
TTD        0.
KE     INT 1,1
PM         11.5
REM    BCD 5H. LANDING ROLL MANEUVER DATA
TSP        0.
TRV        0.
TCH        7.
TBK        0.
ISS        0.
ILR        0
IBS        0
REM    BCD 5I. ENGINE FAILURE STAGE DATA
IC     INT 0,0
XRF1       0.
IT1    INT 0,0
XRF2       0.
IT2    INT 0,0
H1         0.
IH1    INT 0,0
H2         0.
IH2    INT 0,0
HR1        0.
IHR1   INT 0,0
HR2        0.
IHR2   INT 0,0
TR1        0.
ITR1   INT 0,0
TR2        0.
ITR2   INT 0
REM    BCD 5J. BRAKE COND.STAGE DATA
IB     INT 0,0,0
TBK1       1.
IBK1   INT 0,0,0
TBK2       1.
IBK2   INT 0,0,0
REM    BCD 4K. PITCH AUTOPILOT DATA
TST        .2
ALPDL      2.
RFALPH     0.
DELALA     .01
PSH        0.
PSH2       0.
RFALP2     0.
DELQF      8.
DELQTO     0.
DELQL      -10.
DELQU      +25.
DELFD1     5.25
REM    BCD 4L. YAW AUTOPILOT DATA
RFB        .5
DELBA      .05
PSR        10.
DPSIA      .05
RFPSI      .5
PSPSI      0.
DELRL      -20.
DELRU      +20.
REM    BCD 4M. ROLL AUTOPILOT DATA
RFPHI      .5
DPHIA      .05
PSA        -10.
DELPL      -12.
DELPU      +12.
```

```
REM      BCD 5N,THROTTLE AUTOPILOT DATA
TF       INT 0.0
NDF          0.,0.
IR       INT 0,0
NB           1.,1.
NLR          0.,0.
NTO          0.,0.
K2           1.
REM      BCD 40. BRAKE AUTOPILOT DATA
MBC          0.,0.,0.
PD           .1
DELTAW       .001
OMECD1       20.

MBL          0.,0.,0.
MBU          0.,0.,0.
REM      BCD 5P. CONTROL RESPONSE DATA
DELHS        5.25
DELRPD       31.5
DELA         10.
NED1         .1
REM      BCD 40. INITIALIZATION
IAP          3
HR           1.
DELQD        4.56478
DELQDE       4.55478
DELQN        4.55478
DELPD        0.
DELRD        0.
MANLOG       1
PITCHP       1
REM      BCD 2STAGING DATA
REM      BCD 4A.GEARS INTO PROGRAM
INDLG        0
ISTAGE       0
DECRES   BCD 1HR
STESTD       11.0
         TRA
INDLG        -1
REM      BCD 4B.SMOOTH IMPACT STAGE
AINCRS   BCD 3DDELT1DDELT2DDELT3
STEST        -.05,-.05,-.05
         TRA
PRINT        .02
DELTS        .005
AMAXER       .0005
PRTMIN       0.
AINCRS   BCD 1TIMES
STEST        .9
         TRA
ATAB51       7.3516484E-3,1.7027972E-3
ATAB52       -5.2147852E-4,1.0238928E-3
ATAB53       -3.1368631E-4,1.8065268E-5
ATAB56       0.,0.
ATAB57       0.,0.
REM      BCD 5C. EFFICIENT AMAXER STAGE
AINCPS   BCD 4DELTA1DELTA2DELTA3
STEST        .1,.1,.1
         TRA
PRINT        .05
AMAXER       .01
DELTS        .01
REM      BCD 4D.SMOOTH IMPACT STAGE
AINCRS   BCD 1DDELT1
STEST        -.05
         TRA
```

```
      *
PRINT           .02
DELTS           .005
AMAXER          .0005
REM       BCD 5E. EFFICIENT AMAXER STAGE
AINCPS    BCD 1DELTA1
STEST           .1
          TRA
PRINT           .05
AMAXER          .01
DELTS           .01
INDSTF          1
          TRA
```

SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
GENERALIZED COMPUTER PROGRAM

INOSDF 2 CASE LROLL STAGE 1 PAGE 7

A.GEARS INTO PROGRAM

INITIAL PRINT OUT FOR VPCS

XCGRF	AREFF	D1RFF	D2RFF
0.	16.0500000E+02	57.7000000E+00	56.7000000E+00

PRINT CODES IDENTIFYING TIME HISTORY

2SDF

TIME	TIMES	XG77F	YG77F	HGC7F	U777F	V777F	W777F
PI77R	QI77R	RI77R	AMACH	VA77F	DYNPP	XG77F1	YG77F1
ZG77F1	ALPHD	BETAD	ALPHD1	BETAD1	GAM7D	SIG7D	THTPD
FSIPD	PHIFD	AX77F	AY77F	AZ77F	WTR7P	FOC	FCX
FCY	FCZ						

VPCS

AMASS

TFFS

MT	NT					

SAC1

CAVAH	CA	CN	CY	CL	CM	CNN

```
                        SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                            GENERALIZED COMPUTER PROGRAM
                  INDSOF  2      CASE LROLL          STAGE  1      PAGE       6

      2SDF

 0.              0.               0.                0.              10.0000000E+00 29.0274042E+01  0.             59.0569126E+00
 0.              0.               0.            26.5337670E-02 29.6220760E+01 10.4252977E+01 29.6159990E+01   0.
59.9992281E-01 11.4999986E+00     0.                0.              0.            -11.6059853E-01  0.             10.3393988E+00
 0.              0.          57.7454851E-01         0.            -31.6515539E+00 12.4671000E+04  0.              0.
 0.              0.

      VPCS

38.7489900E+02

      TFFS

 0.              0.

                                                                    T(I)
                   0.                  0.

      SAC1

29.7745819E-03-20.8749533E-03 60.9750569E-02  0.                0.                15.3281670E-03  0.

                                                                   FLARE
      AUTS

      ALFDES            PHIDES          TTD

11.5000049E+00  0.                0.

                                                               PITCH AUTOPILOT

      DELON             ALPHAE          ALFHD1          ALPDD1          ALPHET          DELODE

26.8544760E-01-14.0024531E-07  0.                0.              -14.0024531E-07 26.8544760E-01
INTEG RTN.      HT =  1.00000000E-03
      2SDF

 0.              0.               0.                0.              10.0000000E+00 29.0274042E+01  0.             59.0569126E+00
 0.              0.               0.            26.5337670E-02 29.6220760E+01 10.4252977E+01 29.6159990E+01   0.
59.9992281E-01 11.4999986E+00     0.                0.              0.            -11.6059853E-01  0.             10.3393988E+00
 0.              0.          90.1422983E-02         0.            -26.3302709E+00 12.4671000E+04  0.              0.
 0.              0.

      VPCS

38.7489900E+02
```

SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
GENERALIZED COMPUTER PROGRAM

INDSOF 2 CASE LROLL STAGE 1 PAGE 9

```
        TFFS
0.               0.

                   0.                    0.                              T(I)
        SAC1
29.7745819E-03-20.8749533E-03 60.9750569E-02  0.                    0.            15.3281670E-03  0.

      AUTS                                                                  FLARE
      ALPDES           PHIDES           TTD
11.5000000E+00  0.              0.

                                                              PITCH AUTOPILCT
      DELON            ALPHAE           ALFHD1           ALPDD1           ALPHET           DELODE
27.5607208E-01-14.0024531E-07  0.                 0.             -14.0024531E-07 27.5607208E-01

              STAGE ON--DECR. HR
```

SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
GENERALIZED COMPUTER PROGRAM

```
                      INDSDF  2      CASE LROLL        STAGE  1      PAGE     10

              INDLG        -1
              REM    BCD   49.SMOOTH IMPACT STAGE
              AINCRS BCD   30DELT1DDELT2DDELT3
              STEST        -.05 ,-.05,-.05
                     TRA

      ZSDF

 0.             0.             0.             0.             10.0000000E+00 29.0274042E+01 0.             59.0569126E+00
 0.             0.             0.             26.5337670E-02 29.6220760E+01 10.4252977E+01 29.6159990E+01 0.
 59.9992281E-01 11.4999986E+00 0.             0.             0.             -11.6059853E-01 0.             10.3393988E+00
 0.             0.             90.1422983E-02 0.             -26.3302705E+00 12.4671000E+04 0.             0.
 0.             0.

      LGEAR

      DELTA          P              P2             F?             SR             SF             AA             FC2

 0.             35.2800000E+03 0.             0.             31.9713310E-01-38.9549765E+02-78.8562277E+01 0.
 0.             40.3200000E+03 0.             0.             54.9315280E-01-10.7799996E+03-39.6383699E+01 0.
 0.             40.3200000E+03 0.             0.             54.9315280E-01-10.7799396E+03-39.6383699E+01 0.

      MUVP           VGPT           FTRX           FTRY           FTRZ           MA             MB             DDELTA

 33.6000000E-02 29.6159990E+01 0.             0.             -0.            0.             0.             -79.6895154E-01
 33.6000000E-02 29.6159990E+01 0.             0.             -0.            0.             0.             -10.9046626E-01
 33.6000000E-02 29.6159990E+01 0.             0.             -0.            0.             0.             -10.9046626E-01

      SD2            SD1            S              S2D2           S2D1           S2             OMETD1         OMET

 0.             0.             0.             0.             0.             0.             0.             0.
 0.             0.             0.             0.             0.             0.             0.             0.
 0.             0.             0.             0.             0.             0.             0.             0.

      FTRA           FTRB           FTRC           MTX            MTY            MTZ            FXM            FYM

 0.             0.             0.             0.             0.             0.             0.             0.

      FZM            LM             MM             NM

 0.             0.             0.             0.

      VPCS

 38.7489900E+02

      TFFS

 0.             0.
```

```
                     SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                           GENERALIZED COMPUTER PROGRAM
              INOSOF  2      CASE LROLL        STAGE  1      PAGE      11

                                                      T(I)
        0.                 0.
   SAC1
29.7745819E-03-20.8749533E-03 60.9750569E-02  0.           0.           15.3201670E-03  0.
   FLEX
   POINT         XD2F           XD2T           YD2F           YD2T           ZD2F           ZD2T           XD1F
   XD1T          YD1F           YD1T           ZD1F           ZD1T           XD0F           YD0F           ZD0F
10.0000000E-01  0.        91.2652902E-02  0.               0.              0.         -28.9135646E+00  0.
29.0274042E+01  0.         0.             0.          59.0569128E+00  0.          0.              10.5462913E-02
20.0000000E-01  0.        10.2398815E-01  0.               0.              0.         -28.4544208E+00  0.
29.0274042E+01  0.         0.             0.          59.0569128E+00  0.          0.              10.7664860E-02
30.0000000E-01  0.        92.5024062E-02  0.               0.              0.         -26.1584011E+00  0.
29.0274042E+01  0.         0.             0.          59.0569128E+00  0.          0.             -11.9779242E-02
40.0000000E-01  0.        91.5277229E-02  0.               0.              0.         -24.3726223E+00  0.
29.0274042E+01  0.         0.             0.          59.0569126E+00  0.          0.              22.8822210E-02
   AUTS                                                   FLARE
   ALPDES         PHIDES        TTD
11.5000000E+00  0.           0.
                                                  PITCH AUTOPILCT
   DELON          ALPHAE         ALFHO1         ALPDD1         ALPHET         DELCDE
27.5339224E-01-14.0024531E-07  0.             0.          -14.0024531E-07 27.5339224E-01
INTEG RTN.       HT =   8.00000000E-04
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
INTEG RTN.       HT =   1.60000000E-03
```

```
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03

      2SDF

50.0000600E-03 50.0000000E-03 14.8031858E+00  0.                              96.9255346E-01 29.0027313E+01  0.                            59.3321982E+00
 0.            27.2691345E-04  0.             26.5179136E-02 29.6034039E+01 10.4122524E+01 29.5967121E+01  0.
62.9408593E-01 11.5617295E+00  0.             12.6599115E-01  0.            -12.1827675E-01  0.            10.3434516E+00
 0.             0.            94.7837143E-02  0.            -26.7864184E+00 12.4671000E+04  0.             0.
 0.             0.

      LGEAR

      DELTA          P              P2             FT             SR             SF             AA             FCZ

 0.            35.2800000E+03  0.             0.            37.1612726E-01-38.9549765E+02-78.8562277E+01  0.
 0.            40.3200000E+03  0.             0.            44.6574281E-01-10.7799996E+03-39.6383699E+01  0.
 0.            40.3200000E+03  0.             0.            44.6574281E-01-10.7799996E+03-39.6383699E+01  0.

      MUVP           VGPT           FTRX           FTRY           FTRZ           MA             MD             DDELTA
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                        SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                           GENERALIZED COMPUTER PROGRAM
              INDSDF  2      CASE LRULL        STAGE  1      PAGE      13

33.5800000E-02 29.5973017E+01  0.            0.            -0.           0.            0.            -75.5784194E-01
33.6000000E-02 29.5965279E+01  0.            0.            -0.           0.            0.            -98.1007321E-02
33.6000000E-02 29.5965279E+01  0.            0.            -0.           0.            0.            -98.1007321E-02
     SD2            SD1             S            S2D2          S2D1           S2          OMETD1          OMET
 0.             0.             0.            0.            0.            0.            0.            0.
 0.             0.             0.            0.            0.            0.            0.            0.
 0.             0.             0.            0.            0.            0.            0.            0.
     FTRA           FTRB           FTRC          MTX           MTY           MTZ           FXM           FYM
 0.             0.             0.            0.            0.            0.            0.            0.
     FZM            LM             MM            NM
26.4586471E+00  0.            17.8487823E+01 0.
     VPCS
38.7489900E+02
     TFFS
 0.             0.

                0.             0.                          T(I)
     SAC1
38.0142517E-03-21.9773025E-03 62.1249652E-02  0.            0.            12.2575620E-03  0.
     FLEX
     POINT          XD2F           XD2T          YD2F          YD2T          ZD2F          ZD2T          XD1F
     XD1T           YD1F           YD1T          ZD1F          ZD1T          XD0F          YD0F          Z00F
10.0000000E-01  0.            95.6591171E-02  0.            0.           51.4324685E-02-28.3408744E+00  0.
29.0027819E+01  0.             0.           13.1319487E-03 59.2289816E+00  0.            0.           18.5681969E-02
20.0000000E-01  0.            10.4579527E-01  0.            0.           55.1828791E-02-27.5350303E+00  0.
29.0032834E+01  0.             0.           16.3322030E-03 59.2528612E+00  0.            0.           10.7952610E-02
30.0000000E-01  0.            96.6655680E-02  0.            0.          -53.6546557E-02-27.1853834E+00  0.
29.0028375E+01  0.             0.          -15.2076542E-03 59.3247313E+00  0.            0.          -12.0042099E-02
40.0000000E-01  0.            95.9324761E-02  0.            0.           87.3935752E-02-24.3453439E+00  0.
29.0027938E+01  0.             0.           24.9658597E-03 59.4452341E+00  0.            0.           22.9254624E-02

     AUTS                                                  FLANE
```

E-14

```
                                  SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                                      GENERALIZED COMPUTER PROGRAM

                          INDSOF  2      CASE LROLL         STAGE  1      PAGE      14
     ALPDES            PHIDES             TTD
11.5000000E+00  0.              0.

                                                                PITCH AUTOPILCT
     DELON             ALPHAE             ALPHO1             ALPED1             ALPHET             DELODE
28.2700935E-01  61.7294630E-03  12.6599115E-01  0.                  61.7294630E-03  28.2700935E-01

INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
INTEG RTN.        HT =  1.60000000E-03
```

ORIGINAL PAGE IS OF POOR QUALITY

SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
GENERALIZED COMPUTER PROGRAM

INDSOF 2 CASE LROLL STAGE 1 PAGE 15

INTEG RTN. HT = 1.60000000E-03
INTEG RTN. HT = 1.50000000E-03
INTEG RTN. HT = 1.50000000E-03
INTEG RTN. HT = 1.60000000E-03
INTEG RTN. HT = 1.60000000E-03
INTEG RTN. HT = 1.60000000E-03
INTEG RTN. HT = 1.60001000E-03
INTEG RTN. HT = 1.50000000E-03
INTEG RTN. HT = 1.60000000E-03

2SOF

10.0000000E-02	10.0000000E-02	29.5966769E+00	0.	93.7098797E-01	28.9775545E+01	0.	39.6185786E+00
0.	43.2450607E-04	0.	25.5000497E-02	29.5844982E+01	10.3990553E+01	29.5772144E+01	0.
65.6446534E-01	11.6258534E+00	0.	12.9349867E-01	0.	-12.7143277E-01	0.	10.3544256E+00
0.	0.	99.5833827E-02	0.	-27.2744307E+00	12.4671000E+04	0.	0.
0.	0.						

LGEAR

DELTA	P	PZ	FT	SR	SF	AA	FCZ
0.	35.2600000E+03	0.	0.	34.9155442E-01	-38.9549765E+02	-78.8562277E+01	0.
0.	40.3200000E+03	0.	0.	39.9567089E-01	-10.7799996E+03	-39.6303699E+01	0.
0.	40.3200000E+03	0.	0.	39.9567089E-01	-10.7799996E+03	-39.6303699E+01	0.

MUVP	VGPT	FTRX	FTRY	FTRZ	MA	MB	DDELTA
33.6000000E-02	29.5784944E+01	0.	0.	-0.	0.	0.	-72.4166619E-01
33.6000000E-02	29.5801332E+01	0.	0.	-0.	0.	0.	-58.0610413E-02
33.6000000E-02	29.5801332E+01	0.	0.	-0.	0.	0.	-58.0610413E-02

SD2	SD1	S	S2D2	S2D1	S2	OMETD1	OMET
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.
0.	0.	0.	0.	0.	0.	0.	0.

FTRA	FTRB	FTRC	MTX	MTY	MTZ	FXM	FYM
0.	0.	0.	0.	0.	0.	0.	0.

FZM	LM	MM	NM
24.3554817E+00	0.	13.8567145E+01	0.

VPCS

38.7489900E+02

TFFS

```
                         SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                              GENERALIZED COMPUTER PROGRAM
              INOSOF  2     CASE LROLL        STAGE  1     PAGE    16

0.             0.
                                                              T(I)
          0.              0.
    SAC1
38.2764837E-03-23.1194946E-03 63.3356136E-02  0.        0.           90.5418689E-04  0.
    FLEX .
    POINT          XD2F           XD2T           YD2F           YD2T           ZD2F           ZD2T           XD1F
    XD1T           YD1F           YD1T           ZD1F           ZD1T           XD0F           YD0F           ZD0F
10.0000000E-01  0.             10.0200456E-01  0.             0.             86.4210360E-02-27.5360057E+00  0.
28.9776449E+01  0.             0.             48.9324028E-03 59.4615946E+00  0.             0.             18.7159974E-02
20.0000000E-01  0.             10.6787846E-01  0.             0.             37.0598915E-02-28.1584752E+00  0.
29.5795324E+01  0.             0.             42.1630031E-03 59.4914417E+00  0.             0.             18.9455988E-02
30.0000000E-01  0.             10.0935864E-01  0.             0.            -48.1419188E-02-27.6543493E+00  0.
28.9777419E+01  0.             0.            -43.3613657E-03 59.5890238E+00  0.             0.            -12.1519492E-02
40.0000000E-01  0.             10.0557384E-01  0.             0.             90.9801357E-02-2[illegible].2083722E+00  0.
28.9776664E+01  0.             0.             72.5814667E-03 59.0473816E+00  0.             0.             23.1687110E-02
                                                              FLARE
    AUTS
    ALPDES         PHIDES         TTD
11.5000000E+00  0.             0.
                                                     PITCH AUTOPILOT
    DELON          ALPHAE         ALPHD1         ALPDD1         ALPHET         DELCDE
29.0985706E-01 12.5858422E-02 12.9349867E-01  0.             12.5858422E-02 29.0985706E-01
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
INTEG RTN.     HT =  1.60000000E-03
```

SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
GENERALIZED COMPUTER PROGRAM

```
                         INOSDF  2      CASE LROLL        STAGE  1      PAGE      17
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.50000000E-03
INTEG RTN.      HT =  1.60000000E-03
INTEG RTN.      HT =  1.60000000E-03
    2SDF

15.00000000E-02 15.00000000E-02 44.3803598E+00  0.                 90.3652141E-01 28.9520435E+01  0.               59.9059788E+00
 0.             62.8675920E-04  0.             26.4828396E-02  29.5653169E+01 10.3256782E+01 29.5574749E+01  0.
68.1000511E-01 11.6903561E+00  0.             12.8113987E-01   0.            -13.1985334E-01  0.              18.3705821E+00
 0.              0.            10.4422113E-01  0.             -27.7778787E+00 12.4671000E+04  0.               0.
 0.              0.

    LGEAR

    DELTA           P              P2             FT              SR              SF              AA              FCZ

 0.             35.2800000E+03  0.             0.             31.4902661E-01-38.9549765E+02-78.8562277E+01  0.
 0.             40.3200000E+03  0.             0.             37.5811505E-01-10.7799996E+03-39.6383699E+01  0.
```

C-3

```
                         SIX DEGREES OF FREEDOM FLIGHT PATH STUDY
                              GENERALIZED COMPUTER PROGRAM
                 INDSDF  2      CASE LROLL        STAGE  1     PAGE     10
 0.              40.3200000E+03  0.              0.              37.5811505E-01-10.7799995E+03-39.6383699E+01  0.
     MUVP            VGPT             FTRX            FTRY            FTRZ            MA              MB              DDELTA
33.6000000E-02 29.5590914E+01  0.              0.              -0.             0.              0.              -69.1497722E-01
33.6000000E-02 29.5612156E+01  0.              0.              -0.             0.              0.              -24.8322627E-02
33.6000000E-02 29.5612166E+01  0.              0.              -0.             0.              0.              -24.8322627E-02
     SD2             SD1            . S               S2D2            S2D1            S2              OMETD1          OMET
 0.              0.              0.              0.              0.              0.              0.              0.
 0.              0.              0.              0.              0.              0.              0.              0.
 0.              0.              0.              0.              0.              0.              0.              0.
     FTRA            FTRB            FTRC            MTX             MTY             MTZ             FXM             FYM
 0.              0.              0.              0.              0.              0.              0.              0.
     FZM             LM              MM              NM
92.4689028E-01  0.              40.7664789E+00  0.
     VPCS
38.7429900E+02
     TFFS
 0.              0.

                 0.              0.                              T(I)
     SAC1
38.5566271E-03-24.2740905E-03 64.5784391E-02  0.              0.              57.0052216E-04  0.
     FLEX
     POINT           XD2F            XD2T            YD2F            YD2T            ZD2F            ZD2T            XD1F
     XD1T            YD1F            YD1T            Z01F            ZD1T            XD0F            YD0F            ZD0F
10.0600000E-01  0.              10.4805505E-01  0.              0.              86.1109423E-02-27.8883618E+00  0.
28.9521634E+01  0.              0.              93.9243789E-03  59.7316679E+00  0.              0.              19.0731982E-02
20.0000000E-01  0.              10.8968562E-01  0.              0.              77.4745869E-03-28.4993856E+00  0.
28.9533183E+01  0.              0.              52.2996503E-03  59.7377182E+00  0.              0.              11.1880385E-02
30.0000000E-01  0.              10.5236131E-01  0.              0.              -17.7023431E-02-27.8902810E+00  0.
28.9522459E+01  0.              0.              -59.5530950E-03  59.3642718E+00  0.              0.              -12.4153096E-02
40.0000000E-01  0.              10.5231357E-01  0.              0.              74.2456941E-02-26.2991382E+00  0.
28.9521921E+01  0.              0.              11.3863919E-02  60.2231144E+00  0.              0.              23.6383140E-02
```


END
DATE
AUG. 9
1985